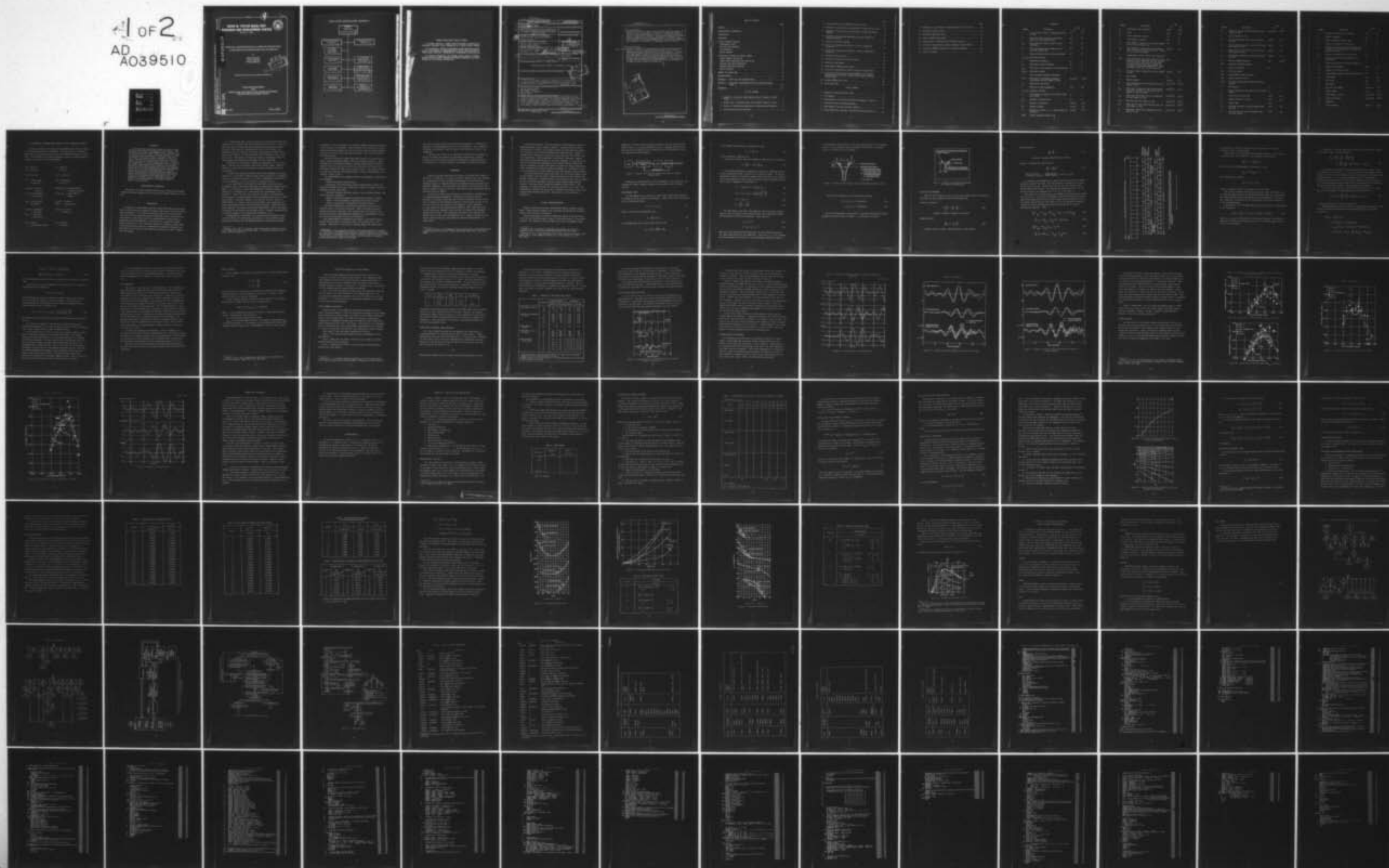
AD-A039 510 DAVID W TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CE--ETC F/G 13/10
STRUCTURAL SEAWORTHINESS DIGITAL COMPUTER PROGRAM ROSAS (A CONV--ETC(U)
MAY 77 S CHUANG, E A SCHROEDER, S WYBRANIEC

UNCLASSIFIED DTNSRDC-77-0001 NL

1 OF 2

AD
A039510



9

Report 77-0001

STRUCTURAL SEAWORTHINESS DIGITAL COMPUTER PROGRAM ROSAS (A CONVERSION FROM SEAWORTHINESS ANALOG COMPUTER)

ADA039510

DAVID W. TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CENTER

Bethesda, Md. 20084

STRUCTURAL SEAWORTHINESS DIGITAL COMPUTER PROGRAM ROSAS (A CONVERSION FROM SEAWORTHINESS ANALOG COMPUTER)

by

Sheng-Lun Chuang
Erwin A. Schroeder
Suzanne Wybraniec

D D C
MAY 17 1977
C

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED

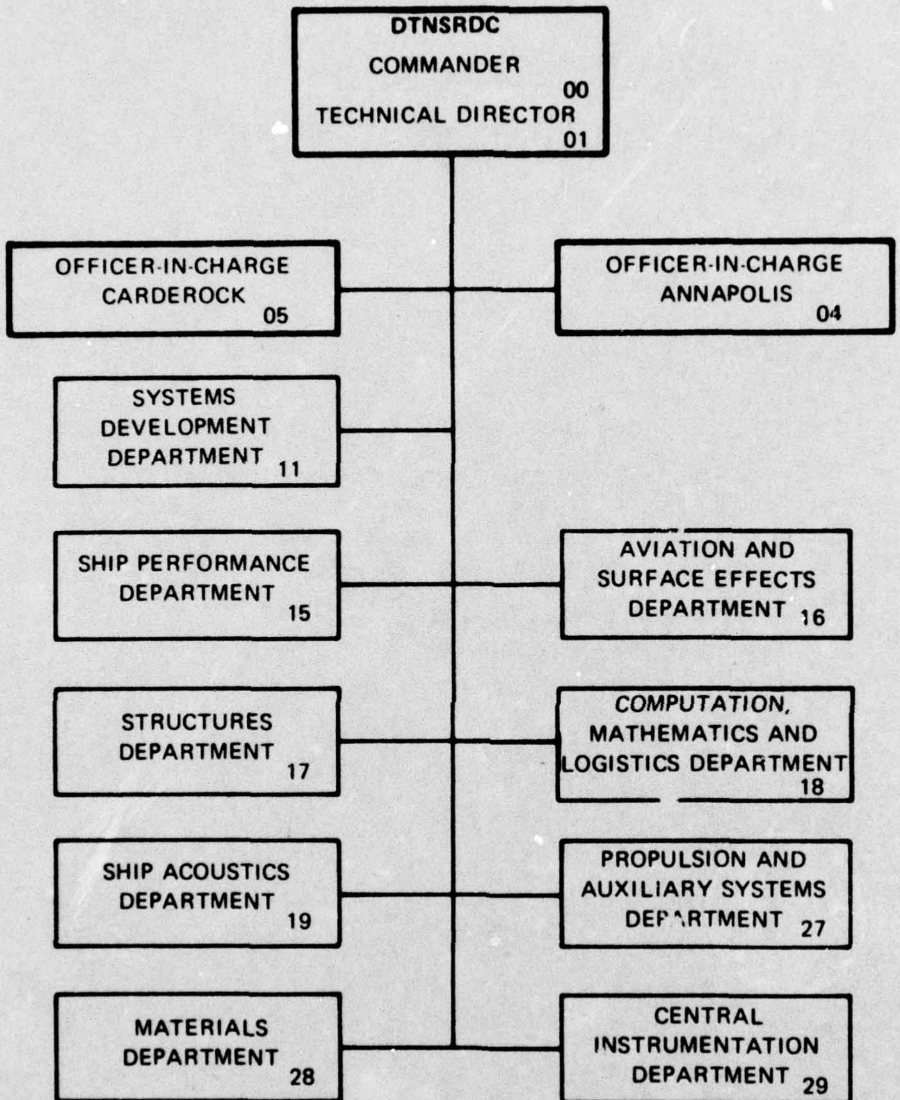
STRUCTURES DEPARTMENT
AND
COMPUTATION, MATHEMATICS, AND LOGISTICS DEPARTMENT
RESEARCH AND DEVELOPMENT REPORT

AD NO.
DDC FILE COPY

May 1977

Report 77-0001

MAJOR DTNSRDC ORGANIZATIONAL COMPONENTS



GPO 905-678

NDW-DTNSRDC 3960/43b (Rev. 11-75)
GPO 928-108

DTNSRDC ISSUES THREE TYPES OF REPORTS

(1) DTNSRDC REPORTS, A FORMAL SERIES PUBLISHING INFORMATION OF PERMANENT TECHNICAL VALUE, DESIGNATED BY A SERIAL REPORT NUMBER.

(2) DEPARTMENTAL REPORTS, A SEMIFORMAL SERIES, RECORDING INFORMATION OF A PRELIMINARY OR TEMPORARY NATURE, OR OF LIMITED INTEREST OR SIGNIFICANCE, CARRYING A DEPARTMENTAL ALPHANUMERIC IDENTIFICATION.

(3) TECHNICAL MEMORANDA, AN INFORMAL SERIES, USUALLY INTERNAL WORKING PAPERS OR DIRECT REPORTS TO SPONSORS, NUMBERED AS TM SERIES REPORTS; NOT FOR GENERAL DISTRIBUTION.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER (14) DTNSRDC [redacted] 77-0001	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) (6) STRUCTURAL SEAWORTHINESS DIGITAL COMPUTER PROGRAM ROSAS (A CONVERSION FROM SEAWORTHINESS ANALOG COMPUTER)		5. TYPE OF REPORT & PERIOD COVERED
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) (10) Sheng-Lun Chuang, Erwin A. Schroeder and Suzanne Wybraniec		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS David W. Taylor Naval Ship Research and Development Center Bethesda, Maryland 20084		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS (17) Task Area SF 43 422 504 Work Unit 1-1730-315
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Sea Systems Command Washington, D.C. 20362 (16) F43422		12. REPORT DATE (11) May [redacted]77
		13. NUMBER OF PAGES 152 (12) 153p.
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) (9) Research and development rept.		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED

DDC
MAY 17 1977

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

There is no warranty, expressed or implied, as to the correctness of the computer program or of its documentation contained in the report. The Government accepts no liability for the results of the program. While a

(Continued on reverse side)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Digital computer program
Ship dynamic response
Ship slamming and whipping
Program ROSAS

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The structural seaworthiness digital computer program ROSAS and users manual are presented in this report. The program was developed using FORTRAN computer language, and it simulates the hull girder structural response of a ship, including dynamic effects when it encounters head seas of the regular, irregular, discrete, standing or other wave forms. Response calculations include the ship rigid and elastic body motion, bending moment, → next page

(Continued on reverse side)

DD FORM 1473 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

387 682

(Block 18 continued)

reasonable effort will be made to answer questions about the program, and a defective copy will be replaced on request, the Government does not undertake to provide maintenance in the future. It should be emphasized that all software, made available to other Government agencies, the private sector for domestic use, to foreign governments, or to business firms having contracts with foreign governments, will be accepted as documented, and no change in or maintenance of documentation shall be required.

(Block 20 continued)

cont

and shear. Vibratory hull girder modes can also be determined, and the effect of bow flare, bottom slamming and springing can be included. Computations have been compared with actual ship responses. The results are in good agreement with those obtained from actual ship sea trials, model experiments, a computer program developed by the Ship Structure Committee, and an earlier analog computer program developed by the David W. Taylor Naval Ship Research and Development Center. The program is a valuable tool for predicting hull girder response of new ship designs or to compare results of model or prototype data.

NTIS White Section
DDC Buff Section
UNANNOUNCED
JUSTIFICATION
BY
DISTRIBUTION/AVAILABILITY CODES
Dist. AVAIL. and/or SPECIAL
A

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOTATION

Symbol	Definition	Unit US	Unit SI
A	Cross sectional area of submerged portion of ship	ft^2	m^2
$\overline{A}$	Cross sectional area due to nonlinear portion of the buoyancy force	ft^2	m^2
A_0	Cross sectional area of ship to still waterline	ft^2	m^2
A_v	Cross sectional area of ship hull for vertical plating only	ft^2	m^2
a	Cross sectional area of structural member	ft^2	m^2
a_1, a_2, a_3	Arbitrary constant		
b	Half-width in general	ft	m
b_1	Ship beam at still waterline	ft	m
b_{1_n}	Ship beam at nth station at still waterline	ft	m
b_2, b_3	Arbitrary constant		
C	Ship structural damping coefficient		
$C(\omega)$	Real part of hydrodynamic damping coefficient per unit length of ship	$ton\text{-}s/ft^2$	$t\text{-}s/m^2$
C_V	Added mass coefficient		
c	Celerity of wave propagation	ft/s	m/s
c_1, c_2, c_3	Arbitrary constant		
d	(1) Distance in general; (2) depth of ship hull girder	ft	m
d_0	Distance of neutral axis	ft	m
E	Modulus of elasticity	ton/ft^2	t/m^2
EI	Bending rigidity	$ton\text{-}ft^2$	$t\text{-}m^2$
G	Modulus of rigidity, i.e., shear modulus of elasticity	ton/ft^2	t/m^2
GBRC	General bending response code		

Symbol	Definition	Unit US	Unit SI
g	Acceleration due to gravity	ft/s^2	m/s^2
Hz	Hertz	cps	cps
h	Double amplitude of wave	ft	m
I	Area moment of inertia	ft^4	m^4
I_0	Area moment of inertia of structural member about its own neutral axis	ft^4	m^4
I_{mz}	Mass moment of inertia about a transverse axis, through its center of mass, of a slice of ship of unit thickness	$ton\text{-}s^2$	$t\text{-}s^2$
J	Longitudinal inertia coefficient (Figure 10)		
KAG	Shear rigidity of ship hull, where K is the numerical factor depending upon the geometry of cross section $K \leq 1$, and A is the cross sectional area of side shell plating and continuous longitudinal bulkhead for carrying shear stress for vertical vibration	ton	t
K_b	buoyancy spring = $\rho g b_1 \Delta x/\Delta x$ per unit length of ship	ton/ft^2	t/m^2
L	Ship length	ft	m
M	Bending moment	ft-ton	m-t
$\bar{m}$	Time varying portion of added mass per unit length of ship	$ton\text{-}s^2/ft^2$	ts^2/m^2
m'_{0_n}	Added mass of ship half section associated with still waterline at nth station for a segment of station length	$ton\text{-}s^2/ft$	ts^2/m
m_0	Added mass associated with still waterline per unit length of ship	$ton\text{-}s^2/ft^2$	ts^2/m^2
m_s, m	Ship mass per unit length of ship	$ton\text{-}s^2/ft^2$	ts^2/m^2
m_V	Added mass of fluid (hydrodynamic mass) per unit length of ship	$ton\text{-}s^2/ft^2$	ts^2/m^2
m_1	Added mass during ship emergence per unit length of ship	$ton\text{-}s^2/ft^2$	ts^2/m^2

Symbol	Definition	Unit US	Unit SI
m_2	Added mass during ship immergence per unit length of ship	ton-s^2/ft^2	ts^2/m^2
n	Station number of ship		
P	Total hydrodynamic force per unit length of ship	ton/ft	t/m
P_1	Inertia force acting on a mass of fluid per unit length of ship	ton/ft	t/m
P_2	Dynamic portion of the buoyancy force per unit length of ship	ton/ft	t/m
P_3	Hydrodynamic damping force per unit length of ship	ton/ft	t/m
p	Bottom slamming pressure	psi	kg/cm^2
RAO	Response amplitude operator		
r	Radius of gyration	ft	m
rms	Root mean square		
SAC	Seaworthiness analog computer		
SCF	Smith correction factor		
SSDC	Structural seaworthiness digital computer program		
T	Wave period	s	s
t	Time coordinate on the ship or in the fluid lamina	s	s
U	Forward velocity of ship	ft/s	m/s
u	Forward velocity of fluid	ft/s	m/s
V	Shear force	ton	t
V_h	Relative horizontal velocity between ship and wave	ft/s	m/s
V_r	Relative vertical velocity between ship and wave surface	ft/s	m/s

Symbol	Definition	Unit US	Unit SI
v	Vertical velocity of wave surface	ft/s	m/s
W	Weight in general	ton	t
X	Space coordinate fixed in ship	ft	m
Y	Vertical translation of ship (↑+)	ft	m
Y_r	Relative vertical translation between ship and sea surface $=Y-Y_w$ = positive when ship moves out of water	ft	m
Y_w	Vertical translation of wave surface	ft	m
y	Waterline distance from baseline, i.e., ship draft	ft	m
β	Ship sectional area coefficient (Figure 8)		
γ	Angular displacement	rad	rad
ΔX	Length of one ship section	ft	m
$\Delta \ell$	Longitudinal length of bottom slamming area	ft	m
Θ_p	Pitch angle	rad	rad
Θ_w	Wave slope	rad	rad
λ	Wavelength	ft	m
μ	Mass per unit length	ton-s^2/ft	ts^2/m
ξ	Impact angle	rad	rad
ρ	Mass density of fluid	ton-s^2/ft^4	ts^2/m^4
ρ_s	Ship mass density	ton-s^2/ft^4	ts^2/m^4
Σ	Summation		
ω	Wave frequency	rad/sec	rad/s

U.S. CUSTOMARY—SI (INTERNATIONAL SYSTEM OF UNITS) CONVERSION FACTORS

The work contained in the report was performed before formal issuance of the metrication policy of the Center. Therefore, the U.S. customary units are used in the report instead of the international system of units (SI). The following list contains conversion factors to enable readers to compute the SI unit values of measurements.

1 ft = 30.48 cm	1 m = 39.37 in
1 in = 25.40 mm	= 3.281 ft
1 in^2 = 6.45 cm^2	1 cm^2 = 0.155 in^2
1 ft^3 = 28.32 liters	1 m^3 = 1000 liters
= 0.02832 m^3	= 35.31 ft^3
1 long ton = 1.016 t	1 newton (N) = (1/g) kg-m/sec^2
= 2240 lb	= (1/9.81) kg (force)
1 lb = 0.454 kg (mass)	1 tonne (t) = 0.984 long ton
= 4.45 N (force)	= 1000 kg
1 psi = 0.0703 kg/cm^2	1 kg/cm^2 = 14.22 psi
= 6.9 kN/m^2	1 N/m^2 = 1 pascal (Pa)
1 ft-lb = 0.1383 kg-m	1 kg-m = 7.23 ft-lb
= 1.356 N-m	= 9.807 N-m
1 Btu = 107.6 kg-m	
= 778.3 ft-lb	
1 hp = 0.746 kw	1 ps = 0.735 kw
= 1.014 metric hp (ps)	= 0.986 hp

ABSTRACT

The structural seaworthiness digital computer program ROSAS and users manual are presented in this report. The program was developed using FORTRAN computer language, and it simulates the hull girder structural response of a ship, including dynamic effects when it encounters head seas of the regular, irregular, discrete, standing or other wave forms. Response calculations include the ship rigid and elastic body motion, bending moment, and shear. Vibratory hull girder modes can also be determined, and the effect of bow flare, bottom slamming and springing can be included. Computations have been compared with actual ship responses. The results are in good agreement with those obtained from actual ship sea trials, model experiments, a computer program developed by the Ship Structure Committee, and an earlier analog computer program developed by the David W. Taylor Naval Ship Research and Development Center. The program is a valuable tool for predicting hull girder response of new ship designs or to compare results of model or prototype data.

ADMINISTRATIVE INFORMATION

This project has been funded and authorized by the Naval Ship Systems Command (035) under Subproject SF 43 422 504, Task 15939, Work Unit 1-1730-315.

INTRODUCTION

The design of a new seagoing ship has usually been based on past experience as well as rules and empirical formulas that are not too involved with higher mathematics. The rules and formulas, prepared by classification societies such as the American Bureau of Shipping, Lloyd's, and others are quite simple to apply and are the only guides needed for ship design by designing offices and shipyards. Such practice is considered reliable because, from year to year, rules and formulas are reviewed, revised, modified, and improved by a group of experienced and reputable engineers and specialists in the fields of ship operation, maintenance, repair, construction, and design.

If a ship design were to deviate from the so-called conventional type, adoption of the standard rules and formulas would be difficult. Therefore, when novel types of ships are being designed, the quasi-static balance method has been adopted to determine hull loads for the design. The technique is to put a ship on a fictitious wave-shaped sea surface and to balance the weight of the ship statically with the buoyancy force of the static wave. Calculations provide ship responses in bending so that the ship designer may determine, accordingly, the ship scantlings.

Calculation by this method is simple but cumbersome, if it is done by hand. With the aid of computers, the process becomes simple. However, a drawback to this method is the omission of dynamic effects of wave and ship motions.

To obtain more reliable information about ship responses, the present tendency is to use either a physical model tested in waves or a mathmetical model for an analytical solution. In solving a mathematical model analytically, one approach is to use an analog or a digital computer.

In the early 1960's, the David W. Taylor Naval Ship Research and Development Center (the Center) developed the seaworthiness analog computer (SAC). It is actually a complicated mathematical simulation model and consists of a sea generator, a ship analog, and a hydrodynamic force generator.[1] Output from the ship analog is fed back to the hydrodynamic force generator to produce dynamic interaction between ship and sea. Computations made to determine responses of an aircraft carrier, the Ex-USS ESSEX (CVA-9), to a specific wave train have been in good agreement with actual measurements made on the ship during sea trials. Unlike the quasi-static approach, this method includes both hydrodynamic effects and dynamic interaction in the analysis and provides as well a realistic representation of the ship response to sea waves during operations at sea.

Although SAC was considered one of the important developments toward realistic analysis in ship design, investment in a large analog computer facility for solving *only* ship response problems could not be justified

---

[1] Andrews, J.N., and S.-L. Chuang, "Seaworthiness Analog Computer," David Taylor Model Basin Report 1829 (Aug 1965). A complete listing of references is given on page 142.

economically. Since the Center has a digital computer facility, utilization of this facility is definitely a logical approach for solving this specific problem because any problem that can be solved by using the analog computer can also be solved by using the digital computer. Thus, the SAC facility never materialized.

Several unsuccessful attempts were made by others to convert SAC to a digital computer program; however, there were numerous obstacles. These obstacles have finally been overcome. Results obtained from the digital computer check very well not only with those obtained from SAC but also with those obtained from model tests and sea trials. This computer program is named the structural seaworthiness digital computer program ROSAS (i.e., response of ship at sea).

The program ROSAS in its present form has the capability of determining the following:

- Ship-hull-bending vibratory modes
- Ship rigid- and elastic-body motions, bending moment, shear, and other hull responses from a ship subjected to regular, irregular, discrete, standing, and other wave forms
- Ship RAO (response amplitude operator), rms (root-mean-square), and other statistical properties
- Effect of bow-flare-slamming in magnifying hull girder response
- Effect of bottom-slamming in magnifying hull girder response
- Effect of springing* in magnifying hull girder response

The listed capabilities are considered sufficient for use in the practical design of ship hull structures. While the present program is limited to head sea conditions, this limitation will be overcome in the near future because three-dimensional mathematical representations of sea and ship are now available but not fully developed.

The program ROSAS is presented in this report. An aircraft carrier (ESSEX) has been chosen to illustrate some of the capabilities of the program. Comparisons of the results of program ROSAS, SAC, model tests,

---

*Springing is a term generally applied to the pseudo-steady-state response of a ship hull in its fundamental, vertical, vibratory mode due to synchronous wave excitation. Springing is often known to generate significant hull stresses in such ships as Great Lakes bulk carriers, etc.; therefore, it requires proper accounting in design of those ships.

sea trials, and other sources are discussed and evaluated. Detailed presentations of program ROSAS are given in the appendixes. Also, methods for obtaining ship parameters used for input to the program are given in the appendixes.

In writing this report, it has been kept in mind that the program can readily be used by the readers for ship-design applications. However, because of the wide range of experience represented by individual users, it is impossible to cover every possible item needed for using the program.

BACKGROUND

A ship can develop appreciable hull stress, associated with transient vibration or "whipping" of the ship in heavy or moderate seas. This whipping may be generated by either emergence and subsidence of the bottom and subsequent impact or nonlinear buoyancy and momentum forces associated with bow-flare immergence. The substantial contributions from bow flareup that induced whipping stresses in the hull girder were vividly demonstrated during rough sea trials of ESSEX.[2]

A theoretical analysis was developed at the Center for investigating the whipping-response phenomenon. The analysis in essence utilized measured or calculated rigid body motion at each transverse section of the ship to compute the instantaneous waterline at each section as well as the velocity of the section relative to the waterline. Next, added mass for each section at each waterline was computed. Then, added mass force at each section was computed as the time rate of change of the momentum imparted by the water. This computation was added to the buoyancy and gravity forces to give total hydrodynamic force. Finally, response of the elastic ship was computed, thus giving the desired bending moments and shear forces.

---

[2] Jasper, N.H. and J.T. Birmingham, "Strains and Motions of USS ESSEX (CVA-9) During Storms Near Cape Horn," David Taylor Model Basin Report 1216 (Aug 1958).

Using this procedure, a detailed analysis of the whipping response of ESSEX was made by using a digital computer.[3] Although there were some discrepancies in details, the maximum stresses were predicted with reasonable accuracy, and the general agreement was considered good. However, changes in hull parameters could not be made easily with this method, and hand computation of the hydrodynamic forces before obtaining a computer solution was undesirable because it was time consuming. In particular, the need for having previous knowledge of ship motions severely limited the choice of mathematical model that would be suitable for design evaluations. These considerations motivated further development of a mathematical model to be computerized for easier handling of the ship dynamic response problem.

The early decision was made on the basis that the analog computer would be more suitable than the digital computer because the analog computer had greater flexibility in changing values of ship parameters and other variables. In fact, the Center possessed a passive analog computer named "network analyzer," that was used in making hull-vibration calculations.[4] Now scrapped, the network analyzer was considered excellent in its time. The abandonment of analyzer has necessarily stimulated conversion of SAC to the program ROSAS mentioned previously. Descriptions of SAC are given in Reference 1.

DIGITAL COMPUTER MODELING

Figure 1 shows the structural seaworthiness digital computer program ROSAS. Three principal elements that constitute the program are the hydrodynamic force, the ship, and the sea.

The sea subroutine is capable of simulating regular sinusoidal waves, a wave train of definite shape or simply a sinusoidal pulse at a prescribed location. The response of the ship feeds back to the hydrodynamic force

---

[3]Andrews, J.N., "A Method for Computing the Response of a Ship to a Transient Force," David Taylor Model Basin Report 1544 (Nov 1963).

[4]McGoldrick, R.T., "Ship Vibration," David Taylor Model Basin Report 1451; Figure 3-8, p. 3-18; Figures A-2 and A-3, p. A-3; Table 8-2, p. 8-10 (Dec 1960).

subroutine to produce dynamic interaction between the ship and the hydrodynamic forces. In case bow emergence occurs, a slamming subroutine computes the bottom slamming forces and adds them to the hydrodynamic forces for computing the ship responses.

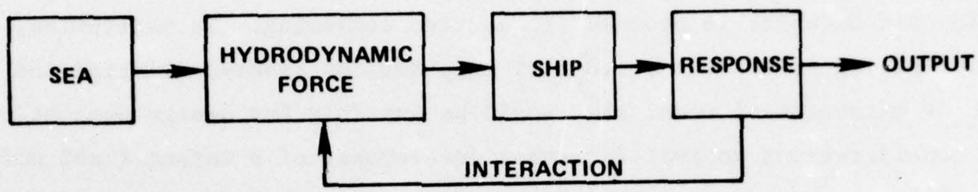


Figure 1 - Diagram of Structural Seaworthiness Digital Computer Program ROSAS

Mathematical representations of the hydrodynamic force, ship, sea, and slamming for digital computer modeling are presented in the following sections.

HYDRODYNAMIC FORCE

The hydrodynamic force consists of three types: added mass; buoyancy, including the Smith correction; and damping. These forces can be represented by the following equations:

$$P = P_1 + P_2 + P_3 \tag{1}$$

where P is the total hydrodynamic force

$$P_1 = \frac{d}{dt} (m_V V_r) \tag{2}$$

is the added mass force or the fluid inertial force

$$P_2 = \rho A \left(g + \frac{dv}{dt} \right) - \rho g A_0 \tag{3}$$

is the dynamic buoyancy force or spring force, and

$$P_3 = -C(\omega) V_r \tag{4}$$

is the hydrodynamic damping force.

The relative vertical velocity between the ship and the sea surface is

$$V_r = \frac{d}{dt}(Y - Y_w) = \frac{d}{dt} Y_r \tag{5}$$

The previous equations are expressed in terms of a coordinate system moving longitudinally with a fixed point in the fluid lamina. If they are expressed in terms of a coordinate system fixed in the ship, these equations can be approximated as follows, neglecting the horizontal component of the fluid velocity (Appendix A of Reference 1):

$$P_1 = -\frac{\partial}{\partial t}(m_V V_r) + U \frac{\partial}{\partial X}(m_V V_r) \tag{6}$$

$$P_2 = \rho g (A - A_0) + \rho A_0 \left(\frac{c}{c + U}\right)^2 \frac{\partial^2 Y_w}{\partial t^2} \tag{7}$$

$$P_3 = -C(\omega) V_r \tag{8}$$

$$V_r = \frac{\partial Y_r}{\partial t} - U \frac{\partial Y_r}{\partial X} \tag{9}$$

The added mass m_V and area A are separated into linear and nonlinear terms so that the effects of the nonlinearities may be examined to assess the importances of hull-form variations. These terms are

$$m_V = m_0 + \bar{m} \tag{10}$$

$$A = A_0 - b_1 Y_r + \bar{A} \tag{11}$$

where m_0 is the added mass associated with the still waterline, and $\bar{m}$ is the time-varying portion of the added mass. The term $b_1 Y_r$ is the rectangular area measured from the still to the instantaneous waterline, where Y_r

is the distance from the still to the actual waterline; $\bar{A}$ is the nonlinear portion of cross sectional area that produces the dynamic or nonlinear portion of buoyancy force; see Figure 2.

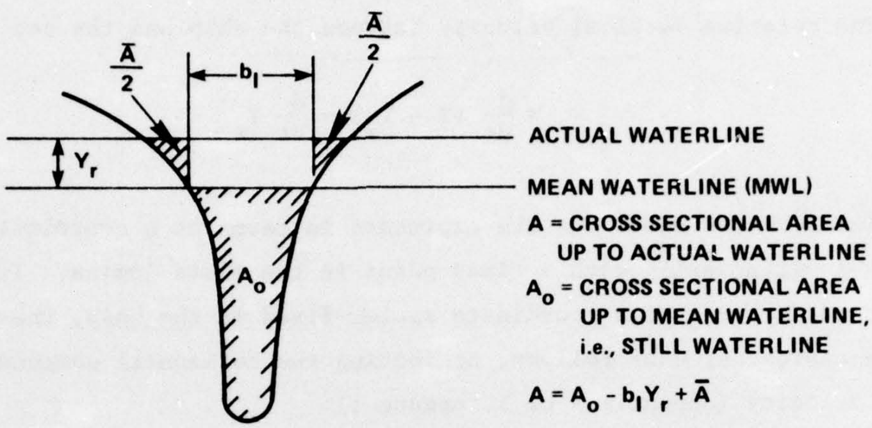


Figure 2 - Method Used to Separate Linear and Nonlinear Buoyancy Forces

The term $\bar{m}$ is defined by the following relationships

$$\bar{m} = m_1 \text{ for } V_r \geq 0 \text{ (emersion)} \tag{12}$$

$$= m_2 \text{ for } V_r < 0 \text{ (immersion)} \tag{13}$$

The two relationships are the result of the added mass being different, depending upon whether ship is immerging or emerging; see Figure 3.

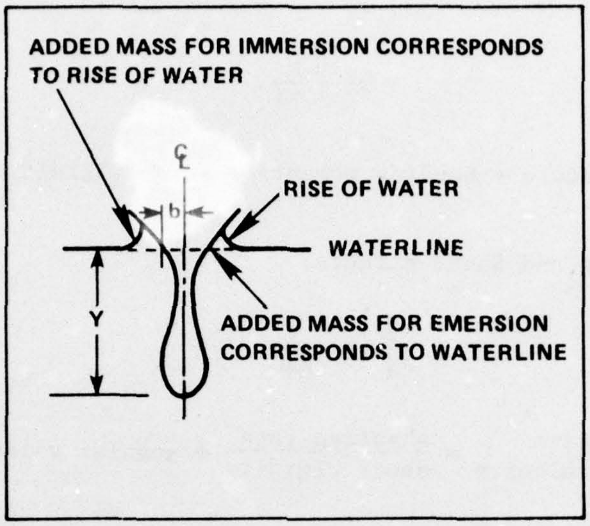


Figure 3 - Method for Determining Added Mass for Immersion and Emersion

SHIP AND SHIP RESPONSE

When the force function P (X, t) acts on the flexible ship, the equations governing the ship response comprise the following set.

Equation of Motion:

$$m \frac{\partial^2 Y}{\partial t^2} + C \frac{\partial Y}{\partial t} + \frac{\partial V}{\partial X} = P \tag{14a}$$

Inertia + damping + shearing = excitation

Moment Equation:

$$\frac{\partial M}{\partial X} = V + I_{mz} \frac{\partial \dot{\gamma}}{\partial t} \tag{14b}$$

Spatial change of moment = shearing force + rotary inertia

Elastic Equation:

$$\frac{\partial \dot{\gamma}}{\partial X} = \frac{\dot{M}}{EI} \tag{14c}$$

Curvature = bending moment/flexural rigidity

Equation of Bending and Shear Effects:

$$\frac{\partial \dot{Y}}{\partial X} = - \frac{\dot{V}}{KAG} + \dot{\gamma} \tag{14d}$$

$$\begin{array}{l}\text{Space derivative} \\ \text{of vertical velocity}\end{array} = \frac{\text{shearing rate}}{\text{shear rigidity}} + \text{angular velocity}$$

To discretize the variable X, 21 equally spaced stations are assigned along the length of the ship. The first station is assigned at the stern and the last at the bow. Twenty half-stations are also assigned, each midway between two stations. The stations are numbered from 0 to 20, and the half stations are numbered from 0.5 to 19.5; see Figure 4.

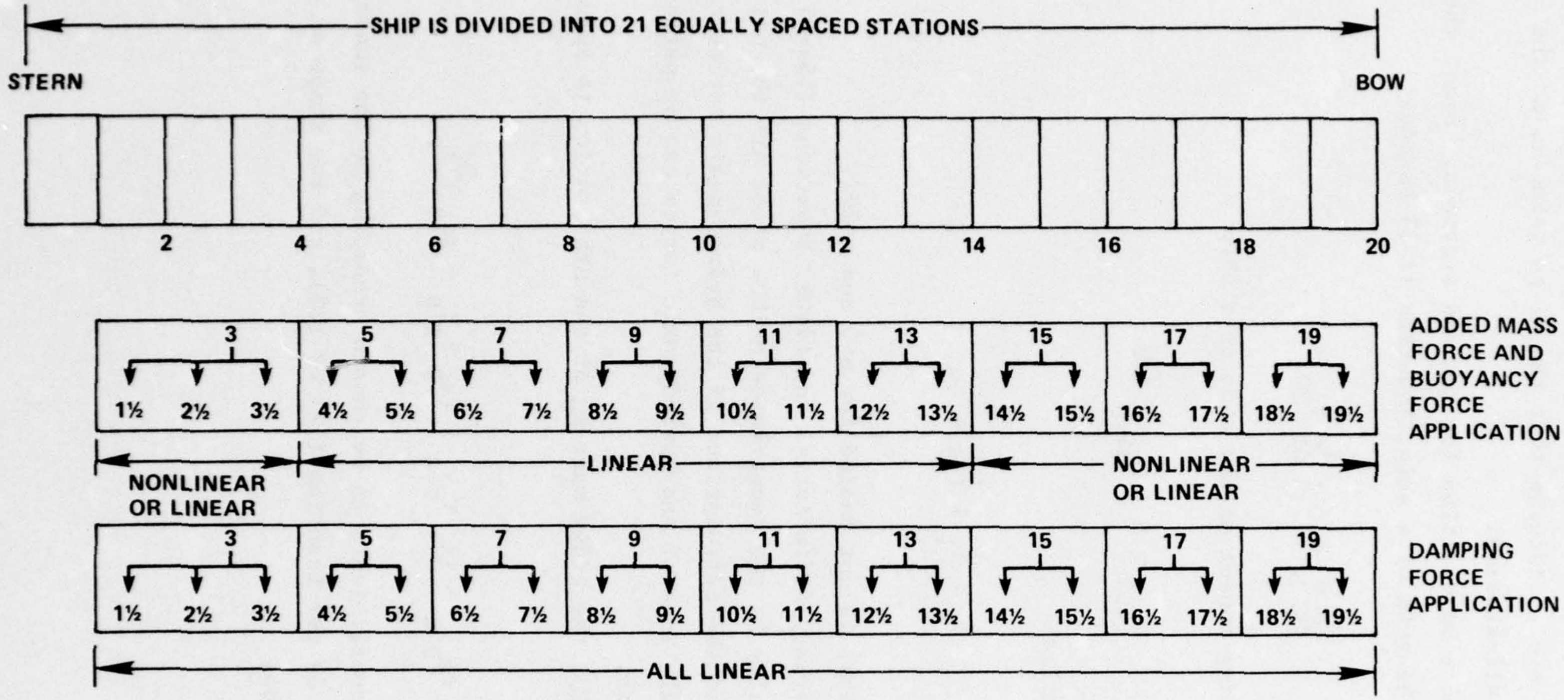
The values of $\dot{\gamma}$, V, *KAG, and* I_{mz} *are lumped at the stations*; the values of $\dot{Y}$, M, EI, m, C, and force P are lumped at the half-stations. With these quantities lumped and with the distance between stations denoted by ΔX, we can replace the derivatives with respect to X in the system of Equations (14a-d) by central difference quotients to get a system of ordinary differential equations. Thus,

$$\frac{d}{dt} \dot{Y}_{n+\frac{1}{2}} = (P_{n+\frac{1}{2}} - (C\dot{Y})_{n+\frac{1}{2}} - (V_{n+1} - V_n)/\Delta X)/m_{n+\frac{1}{2}} \tag{15a}$$

$$\frac{d}{dt} \dot{\gamma}_n = ((M_{n-\frac{1}{2}} - M_{n+\frac{1}{2}})/\Delta X - V_n)/(I_{mz})_n \tag{15b}$$

$$\frac{d}{dt} M_{n+\frac{1}{2}} = EI_{n+\frac{1}{2}} (\dot{\gamma}_{n+1} - \dot{\gamma}_n)/\Delta X \tag{15c}$$

$$\frac{d}{dt} V_n = KAG_n (\dot{\gamma}_n - (\dot{Y}_{n+\frac{1}{2}} - \dot{Y}_{n-\frac{1}{2}})/\Delta X) \tag{15d}$$



NOTE: AT STERN, HYDRODYNAMIC FORCES DISTRIBUTED EQUALLY TO STATIONS 1½, 2½, AND 3½. AT OTHER STATIONS, HYDRODYNAMIC FORCES DISTRIBUTED EQUALLY TO TWO ADJACENT HALF STATIONS.

Figure 4 - Ship Stations Used in Analysis

The subscripts n, $n+\frac{1}{2}$, etc., indicate that the given value is at the indicated station or half-station.

These equations must be modified for the end stations. Since there is no shearing force at the ends of a ship, Equation (15d) becomes

$$\frac{d}{dt} V_0 = 0 \ ; \ \frac{d}{dt} V_{20} = 0$$

Also, there is no bending moment at the ends of a ship

$$M_{0.5} \approx 0 \ ; \ M_{19.5} \approx 0$$

thus Equation (15c) implies

$$\dot{\gamma}_0 \approx \dot{\gamma}_1 \ ; \ \dot{\gamma}_{20} \approx \dot{\gamma}_{19}$$

Therefore, Equation (15b) is not needed for n=0 and n=20.

The system of ordinary differential equations, Equations (15a-d), can be integrated numerically by the Runge-Kutta method after all the parameters have been evaluated and the calculations of the hydrodynamic forces have been completed. Calculations of the hydrodynamic forces can be performed in the following manner.

From Equation (10), the added mass m_V at the nth station is approximated by the expression

$$(m_V)_n = (m_0)_n + (a_1 \ Y_r)_n + (a_2 \ Y_r^2)_n + (a_3 \ Y_r^3)_n \tag{16}$$

where a_1, a_2, a_3 are coefficients to be chosen according to the emersion or immersion of the ship as per Equation (12) or (13), and the shape of ship section at that station.

In Equations (5) and (9), the vertical velocity at nth station relative to the water surface is approximated by

$$(V_r)_n \approx \left(\frac{\partial}{\partial t} Y_r\right)_n - U\left(\frac{\partial}{\partial X} Y_r\right)_n$$

$$\approx \dot{Y}_n - \left(\frac{\partial}{\partial t} Y_w\right)_n - \frac{U}{4\Delta X}\left((Y_r)_{n+2} - (Y_r)_{n-2}\right) \tag{17}$$

The relative displacement of the moving waterline with respect to the still waterline is given by $Y_r = Y - Y_w$. The vertical displacement of water surface Y_w and its vertical velocity $\partial Y_w/\partial t$ are provided by a subroutine that simulates the sea. The vertical displacement of ship Y is obtained by adding the equation $dY/dt = \dot{Y}$ to the system of Equations (14). The sign of Y_r is taken as positive when the ship at nth station moves out of the water and away from the water surface. The ship speed is prescribed.

At Station 3 Equation (17) is modified, since for $(Y_r)_1 = Y_1 - (Y_w)_1$, $(Y_w)_1$ is not available. The modified equation for Station 3 is

$$(V_r)_3 \approx \dot{Y}_3 - \left(\frac{\partial}{\partial t} Y_w\right)_3 - \frac{U}{2\Delta X}\left((Y_r)_5 - (Y_r)_3\right) \tag{18}$$

A similar modification is made for Station 19.

Combining Equations (2), (6), and (17), the hydrodynamic inertial force due to the change in momentum of the added mass $\frac{d}{dt}(m_V V_r)$ at the nth station is

$$(P_1)_n = -\frac{d}{dt}(m_V V_r)_n$$

$$= -\frac{\partial}{\partial t}(m_V V_r)_n + U\frac{\partial}{\partial X}(m_V V_r)_n$$

$$= -(m_V \frac{\partial}{\partial t} V_r)_n - (V_r \frac{\partial}{\partial t} m_V)_n + U\frac{\partial}{\partial X}(m_V V_r)_n$$

$$\approx -(m_V)_n\left(\ddot{Y}_n - (\ddot{Y}_w)_n - \frac{U}{4\Delta X}\left((\dot{Y}_r)_{n+2} - (\dot{Y}_r)_{n-2}\right)\right)$$

$$- (V_r)_n \, (a_1 + 2a_2 \, Y_r + 3a_3 \, Y_r^2)_n \, (\dot{Y}_r)_n$$

$$+ \frac{U}{4\Delta X} \left((m_V \, V_r)_{n+2} - (m_V \, V_r)_{n-2} \right) \tag{19}$$

with the appropriate modifications for Stations 3 and 19 similar to Equation (18).

From Equation (11), the submerged cross sectional area at a station is approximated by the expression

$$A = A_0 - b_1 \, Y_r + b_2 \, Y_r^2$$

The coefficients b_1 and b_2 are chosen according to the vertical displacement of the station relative to the water surface. Substituting the previous expression into Equation (7), we obtain the expression for the dynamic buoyancy force as follows, i.e.,

$$P_2 = \rho g \, (- b_1 \, Y_r + b_2 \, Y_r^2) + \rho A_0 \left(\frac{c}{c + U} \right)^2 \frac{\partial^2 Y_w}{\partial t^2} \tag{20}$$

The values of c and $\partial^2 Y_w / \partial t^2$ are provided by the subroutine that simulates the sea.

The terms $m \, \ddot{Y}$ in Equation (14a) and $m_V \, \ddot{Y}$ in Equation (19) can be transposed by adding m_V to m to form the term $(m + m_V) \, \ddot{Y}$ as part of Equation (14a), since Equation (19) is a part of Equation (14a).

At the present state of the art, there is no accurate method to determine the damping coefficients for structures and fluids. The complex structural and load distribution of a ship make determination of damping coefficients cumbersome. The best results that may be expected will show good correlation between calculated and test-determined values for the fundamental mode of ship hull only. This is also true for fluid damping. Fortunately, the inaccuracy of the damping coefficient will not much affect the maximum magnitude and frequency of the fundamental mode between the computer model and the actual ship. Therefore, only an approximation method will be adopted for the present computer program. This of course can easily be improved later when a more accurate method is available.

The total hydrodynamic force is then the sum of P_1 given by Equation (19) for inertial force, P_2 given by Equation (20) for buoyancy force, and P_3 given by Equation (4) for hydrodynamic damping. Information about the total hydrodynamic force is needed to integrate the system of Equations (15).

SEA GENERATION

Three types of seas are used for computer input. They are simple harmonic excitations, sinusoidal seas, and a specific discrete wave train. Simple harmonic excitation is a sinusoidal exciting force applied at a prescribed location of the ship, e.g., Station 10. Therefore, it does not represent an actual sea condition but is used to determine frequencies of the hull girder at various modes. The sinusoidal sea is a sinusoidal wave train moving without change in form and at a constant wave velocity from the bow to the stern of the moving ship. This is approximated by a delay function that caused the wave train to appear at points along the ship with a time delay equal to the distance from the bow, divided by the sum of ship speed and wave velocity. Sinusoidal sea excitation permits determination of RAO, an important tool for statistical analysis in ship design. The discrete wave train is composed of a series of sinusoidal waves by superposition to curve fit and wave data from the sea trials. This is used to verify dependability of program ROSAS by comparing computer output with sea-trial test results.

No random sea excitation was employed. If this is needed, it is necessary to start with statistical representation of a random sea and then to convert it to random sea excitation in the time domain. From the random sea excitation, the ship responses can be obtained and be converted into the statistical representation for the lifetime prediction of the ship.

The same results can be obtained with response amplitude operators. Of course, this is a more direct and shorter process than the method given previously.

BOTTOM SLAMMING

Bottom slamming occurs when the ship bottom is on or above water during immersion, i.e.,

$$\left.\begin{array}{l}(1)\ Y_r \geq y \\ \\ (2)\ V_r \leq 0\end{array}\right\} \qquad (21)$$

Both conditions (1) and (2) must be satisfied to generate bottom slamming. Slamming starts at a location where $Y_r=y$, and $V_r=0$. It ends at a time where $Y_r=y$ with $V_r<0$, and $Y_r<y$ elsewhere.

At any t during slamming, the impact area can generally be assumed to be triangular in shape, and the load is estimated to be

$$\text{Load} = \Sigma p = \frac{1}{2}\, p_{max}\ \Delta\ell\ b_1\ (144/2240)$$

where $\Delta\ell$ is the longitudinal distance from keel to bilge where the water surface intersects with the ship bottom.

b_1 is the width of ship bottom at impact

p_{max} is the maximum impact pressure. The method for determining p_{max} is given in Reference 5. The slamming load is proportionally added to two adjacent stations together with the hydrodynamic forces for ship response.

---

[5] Chuang, S.-L. et al., "Experimental Investigation of Catamaran Cross-Structure Slamming," NSRDC Report 4653 (Sep 1975).

DISCUSSION OF RESULTS BY DIGITAL METHOD

Since the program ROSAS has newly been converted from SAC, it is necessary to verify its feasibility and accuracy. For comparison, ROSAS was programed with three excitations the same as those used by SAC, i.e., simple harmonic excitation, sinusoidal seas, and discrete wave trains. Slamming response is added to the program. In addition, RAO's for the ship are also obtained and are compared with other methods.

Since sea trials were made with ESSEX, which was simulated on SAC, the ship was used to test the performance of the program ROSAS also. The structural and hydrodynamic parameters that describe ESSEX are taken from Reference 1. To facilitate the use of the program, Appendix A is provided for determining these parameters.

SIMPLE HARMONIC EXCITATION

Vibration modes of the ESSEX hull were observed during sea trials. Thus it has been possible to check the program by comparing vertical bending modes obtained from simple harmonic excitations of the computer model with those of the full-scale ship. The tests also consist of comparing the frequencies of fundamental bending modes by ROSAS with those by GBRC (the general bending response code).[6] and by SAC. The frequency of a fundamental mode was determined using the program ROSAS by searching for the frequency of a point of sinusoidal exciting force that produced the largest response.

Three types of test were made, represented as follows:

Case 1 - Only ship structural mass was used; no hydrodynamic forces were applied.

Case 2 - Added mass was added to structural mass; however, no hydrodynamic buoyancy force was applied.

Case 3 - Added mass was added to structural mass, and hydrodynamic buoyancy force was applied.

---

[6] Henderson, F., "Transient Response Calculation in the Frequency Domain with General Bending Response Program (GBRP)," NSRDC Report 3613 (Feb 1971).

Neither structural nor hydrodynamic damping has been added for the three cases because the damping force does not influence or affect the frequency of fundamental mode very much in actual hull vibration of the ship.

Analog computer results are available only for Cases 2 and 3; sea trial results also are available for Case 3. For Case 3 only the linear part of the buoyancy force could be simulated with GBRC. For all three cases, the ship speed was zero, and calm seas were used. Frequencies in hertz for the fundamental mode of the ship hull obtained from various methods are compared as follows:

Case	ROSAS	GBRC	SAC	Sea Trials
1	1.043 Hz	1.043	-	-
2	0.742	0.750	0.734	-
3	0.749	0.763	0.744	0.825

Frequencies determined by digital simulation of ROSAS agree with those determined by the other simulations within 2 percent. However, the frequency determined by ROSAS is about 91 percent of that obtained from the sea trial. The value of 0.825 Hz was calculated from the record given in Figure 5a of Reference 2. This discrepancy in frequency is considered reasonable for full-scale measurement at sea.

STEADY-STATE SINUSOIDAL WAVE EXCITATION

The actual regular sea wave form can generally be represented to a close approximation by a sinusoidal wave, which is much easier to handle mathematically than the usual approximation by a trochoidal form. Therefore, for this type of test, the sea was represented by a sinusoidal wave train moving at a constant wave velocity from the bow to the stern of the ship. The theoretical wave velocity c is

$$c = g/\omega$$

which generally agrees with the values obtained from observations at sea.

The test results for this type of excitation were available only in Reference 1, i.e., by SAC. Unfortunately, only the linear analysis was performed for this type of excitation. The ship responses reported in Reference 1 were the vertical displacement of the ship at a station relative to the surface of the sea, the pitch angle of the ship, the hydrodynamic force acting at a station, and the bending moment at a station. The test results for Case 6 of Reference 1 and ROSAS are compared in Table 1. In this case, 0.35 rad/s of the wave frequency and 16 knots of the ship speed were programed.

TABLE 1 - RESPONSE TO SINUSOIDAL WAVE TRAIN*

	Station**	Phase Angle***		Amplitude	
		Digital	Analog	Digital	Analog
Wave Height in Feet	3	147	145	10	9.74
	11	71	80	10	9.97
	19	0	0	10	9.88
Relative Displacement in Feet	3	167	114	1.53	2.29
	7	247	283	3.04	1.52
	11	230	270	2.42	1.21
	15	107	87	2.72	2.58
	17	84	79	5.82	5.91
	19	72	74	9.65	9.81
Pitch Angle in Degrees	10.5	9	3	2.02	1.97
Hydrodynamic Force in Tons	3	317	285	162	281
	15	267	242	289	225
	17	248	242	289	275
	19	239	239	135	135
Bending Moment in Foot-Tons	4	247	275	$23.9 \cdot 10^3$	$37.6 \cdot 10^3$
	8	252	263	$104 \cdot 10^3$	$126 \cdot 10^3$
	10	268	257	$121 \cdot 10^3$	$135 \cdot 10^3$
	12	244	252	$112 \cdot 10^3$	$116 \cdot 10^3$
	16	237	240	$37.5 \cdot 10^3$	$35.2 \cdot 10^3$

*Linear analysis: 16 knots of ship speed, 0.35 rad/s of wave frequency.
**Station 0 is at stern; Station 20, at bow.
***Angle given is the relative phase of the measured quantities with respect to wave position at the bow.

The values obtained from ROSAS show trends similar to the values reported for SAC and are of the same order of magnitude. A trend pointed out in Reference 1, and apparent in ROSAS results, is that the relative displacement is large at the bow and small at the stern, and the phase angle of the relative displacement at the stern is close to that of the wave.

The steady-state sinusoidal wave excitation is a very important tool for analyzing ship responses. Further application of this method will be discussed and evaluated in the section about response amplitude operators.

DISCRETE WAVE-TRAIN EXCITATION

For this kind of test, the surface of the sea was represented by an approximation to a discrete wave train recorded during ESSEX sea trials; see Figure 5.[2] Wave height, pitch angle, and midship-bending stress were recorded at the same time. The discrete wave record of the sea trials was so selected that the ship produced whipping of the hull girder.

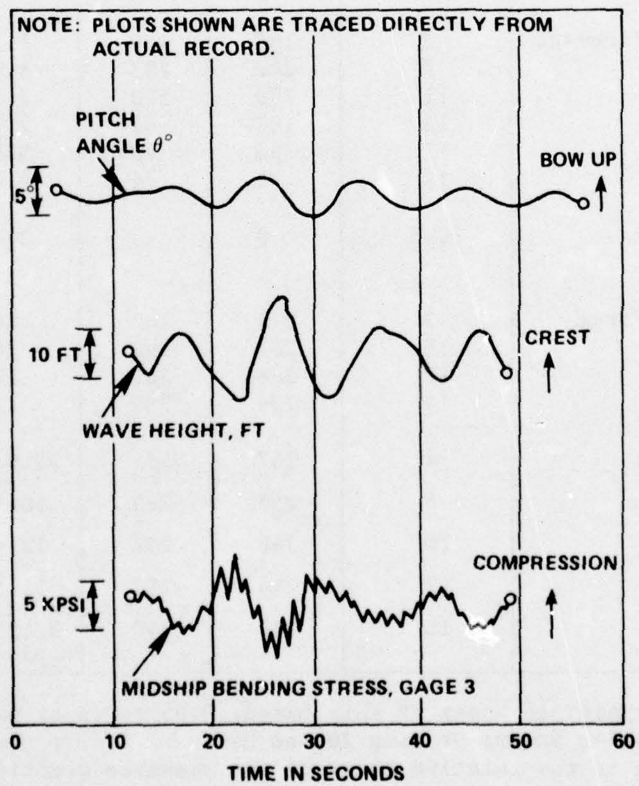


Figure 5 - Actual Records for Ex-ESSEX (CVA-9) Sea Trials (Figure 5a, Reference 2)

The approximated wave heights for both ROSAS and SAC were smoothly connected to a long period of steady sinusoidal waves to avoid extraneous whipping excitation. Thus, the approximations had the same general shape as the recorded large waves but were different in details.

Figure 6 compares wave heights, pitch angles, and midship bending moments obtained from ROSAS and sea trials; Figure 7 compares similar results obtained from ROSAS and SAC. As expected, the ship responses recorded during the sea trials and computed by ROSAS and SAC are similar in character but different in details. The characteristic common to all is that the large discrete wave generated large bow immersions to excite whipping of the hull girder. Results show good agreement in maximum magnitudes among them in ship responses. However, in the sea trial data, a previous excited whipping was not quite damped out before the next excitation began. Since the damping coefficient used in the digital program was not determined from an actual ship, the excited whippings were damped out at different rates for the ship and the computer model.

It was evident that the ship was excited when the bow was pitched steeply down to the high wave. Whipping was obviously due to the pronounced bow flare of the ship section, resulting in an impulse at the ship bow. Bow flare is not the same as bottom slamming, even though both types of impulse would generate whippings of the hull girder. Bottom slamming is more common than bow flare for most ships, and ROSAS in its present form is also programed for bottom slamming, which will be discussed later.

RESPONSE AMPLITUDE OPERATORS

The irregular sea surface may be represented by the sum of a great number of small-amplitude sine waves having different directions and periods. (Wave length and period have a fixed relation.) Ship response to an irregular seaway may be represented by the sum of the ship responses to the simple sine wave components. This linear superposition of random wave theory is known as the energy spectrum analysis and applies not only for ship responses to random sea waves but also to any form of energy such as heat, electricity, light, sound, and many mechanical phenomena.

Figure 6 - Response to Discrete Wave Train-Digital Program and Sea Trial

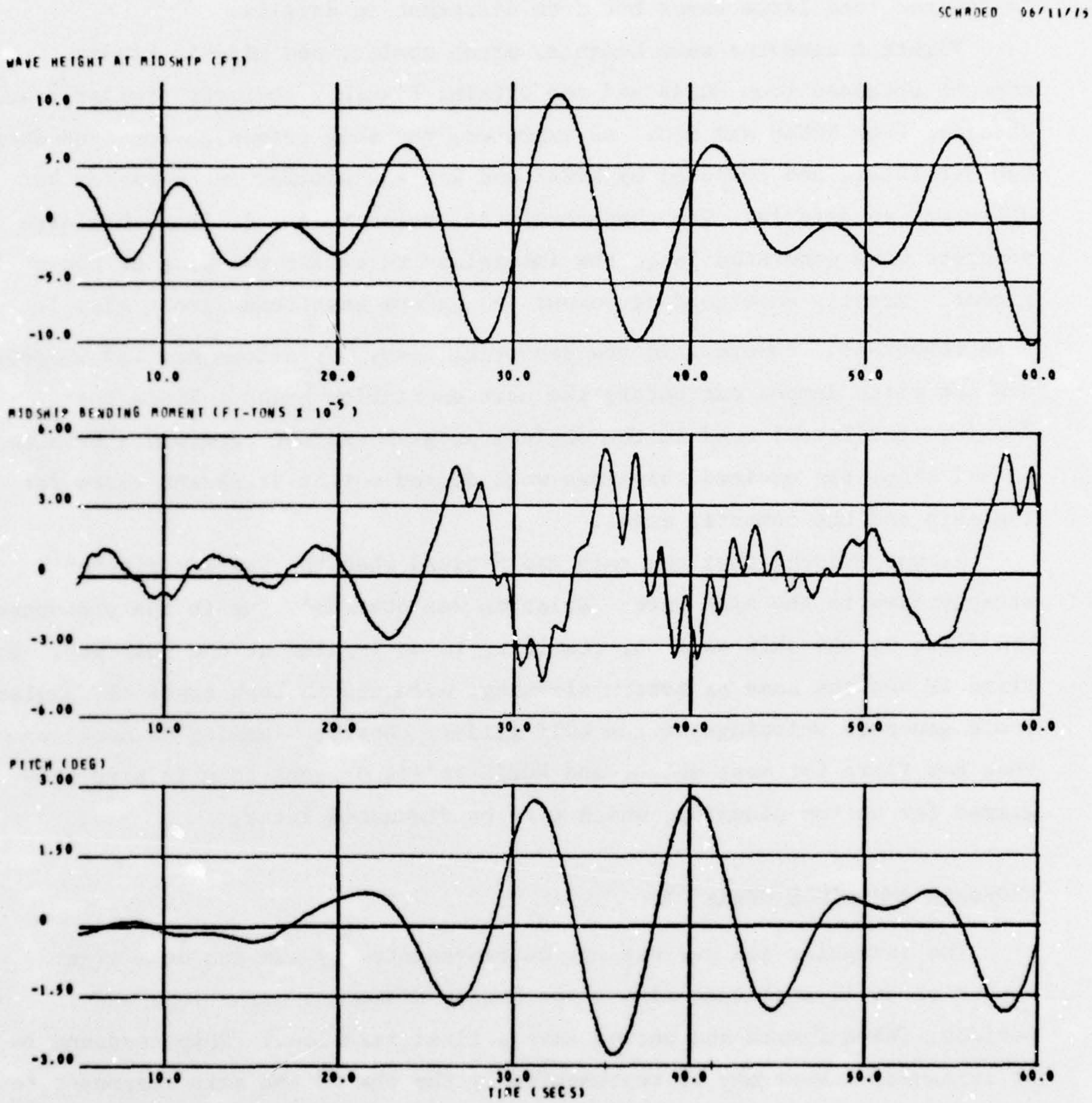


Figure 6a - Actual Output from Program ROSAS

Figure 6 (Continued)

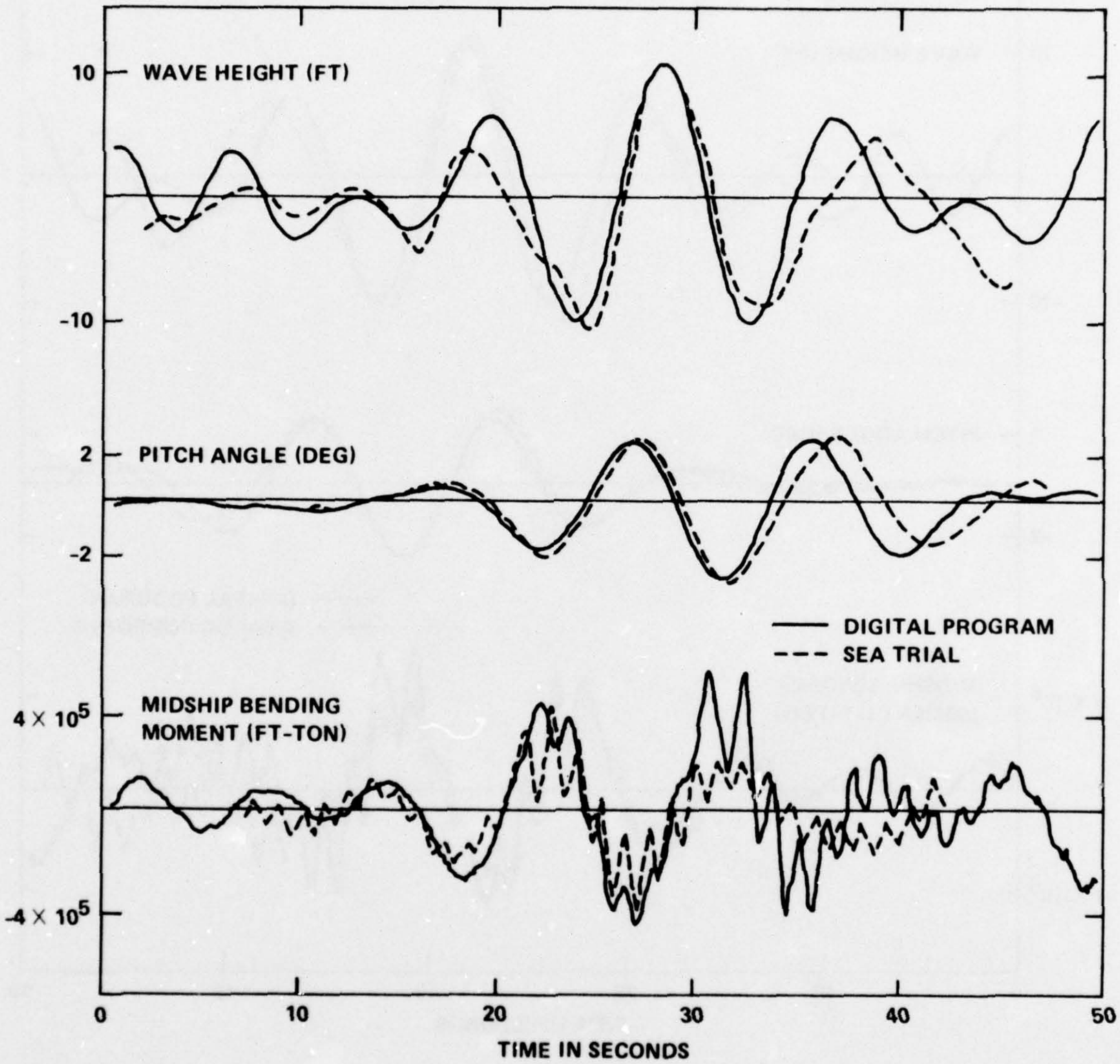


Figure 6b - Comparisons Between ROSAS Output and Sea Trial Data

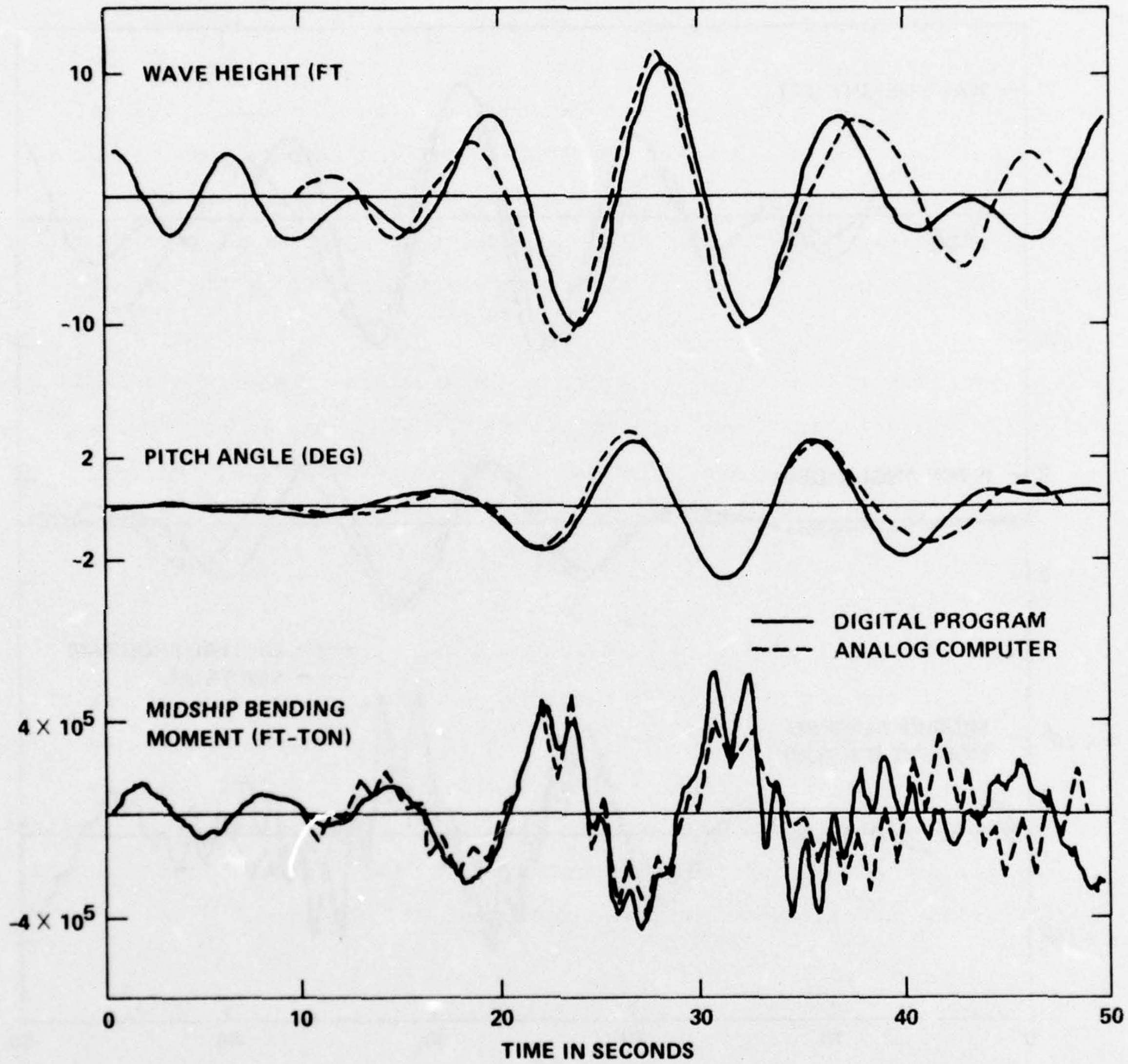


Figure 7 - Response to Discrete Wave Train-Digital Program and Analog Computer

Even though the theory seems to be complex, it may comfort the naval architect and ship designer to know that this powerful tool can be applied in ship design without a detailed understanding of the unusual mathematics. Any seaway can be characterized simply by a "wave energy spectrum." When a squared RAO curve is multiplied by the wave spectrum, a ship response spectrum is obtained. This can be pitch, heave, bending moment, or any other ship response. To obtain the response amplitude operator: First, apply one sine wave at a time to the ship to get the ship response. Second, divide each ship response by the single sine wave; the result is RAO. This operation usually requires the physical model test under regular or irregular sea conditions; now, however, it can be obtained directly by the program ROSAS.

Figure 8 compares RAO's for pitch angle and bending moment of ESSEX among the physical model tests,[7] the program SCORES,[7] and ROSAS results. Very good agreement is shown. The SAC is capable of obtaining RAO's. However, it was not included in Reference 1.

BOTTOM SLAMMING

The ship chosen for the present example has deep draft, and its bow does not come out of the water at all. As an illustration, the ship draft was drastically reduced in the program so that bottom slamming occurred during ship operations. Figure 9 shows bottom slamming output from the computer; nonlinear terms have been omitted in the program to avoid bow-flare slamming. Since no actual bottom-slamming data have been obtained from the sea trials of ESSEX, no comparison can be made at this time.

---

[7] Kaplan, P. et al., "An Investigation of the Utility of Computer Simulation to Predict Ship Structural Response in Waves," Ship Structure Committee Report SSC-197 (Jun 1969).

Figure 8 - Comparisons of Response Amplitude Operators Obtained by Different Methods

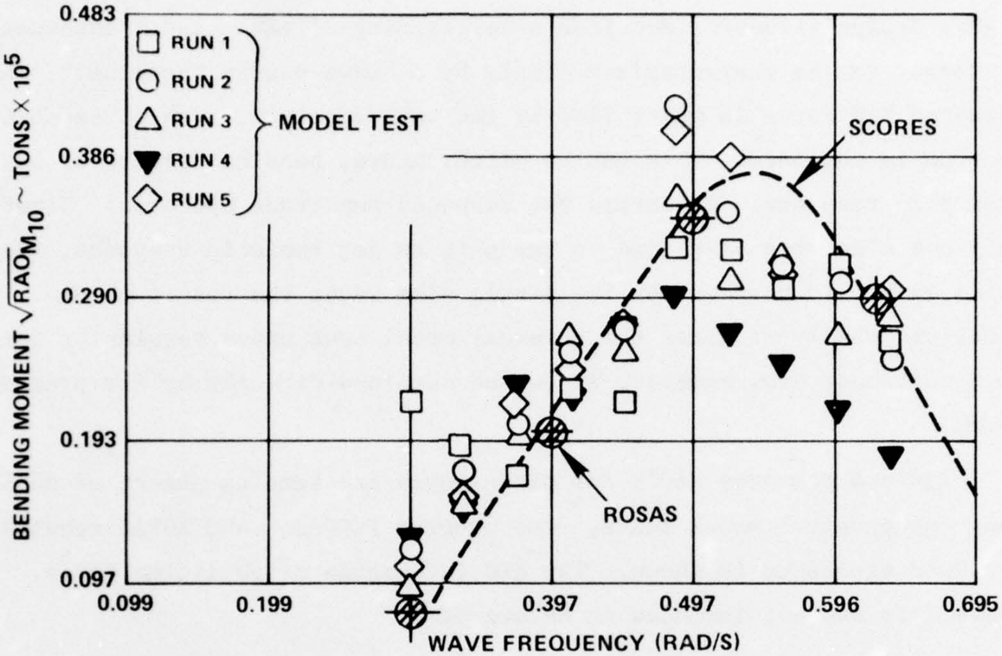


Figure 8a - Square Root of Bending Moment $RAO_{M_{10}}$, Zero Speed

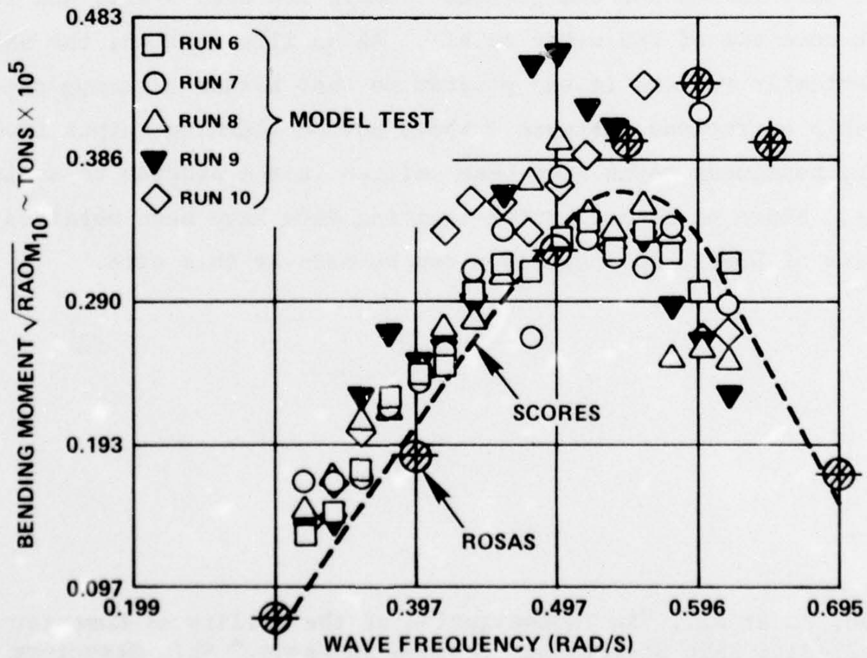


Figure 8b - Square Root of Bending Moment $RAO_{M_{10}}$, 13.8 Knots

Figure 8 (Continued)

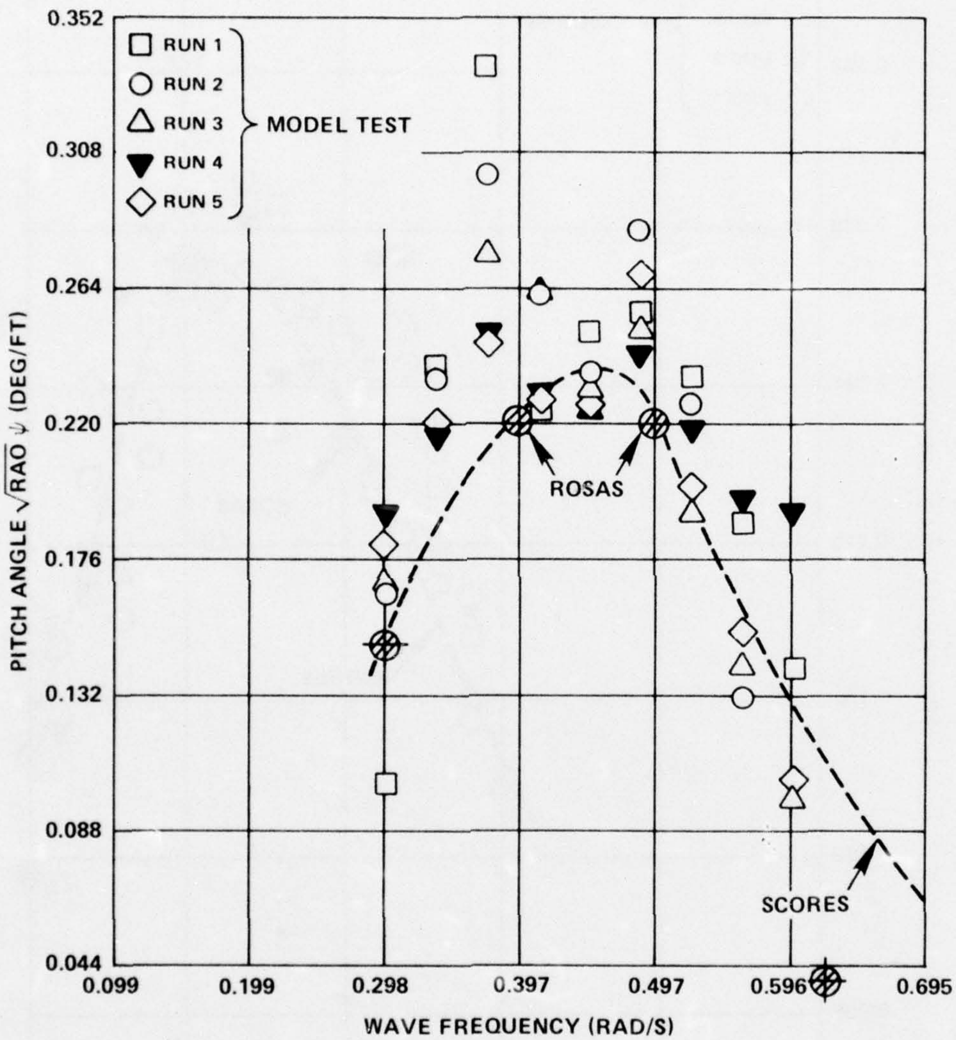


Figure 8c - Square Root of Pitch Angle RAO_{ψ}, Zero Speed

Figure 8 (Continued)

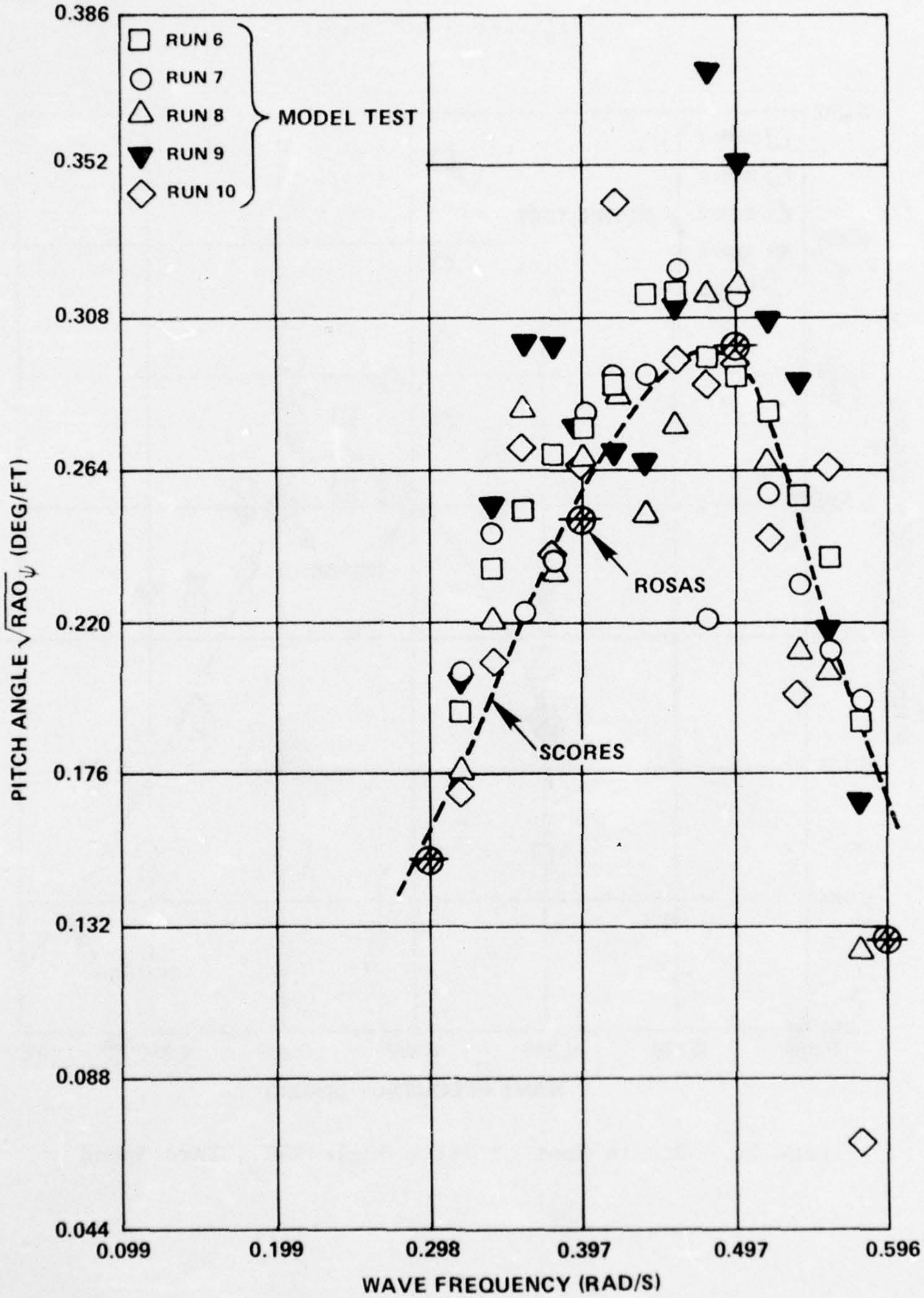


Figure 8d - Square Root of Pitch Angle RAO_ψ, 13.8 Knots

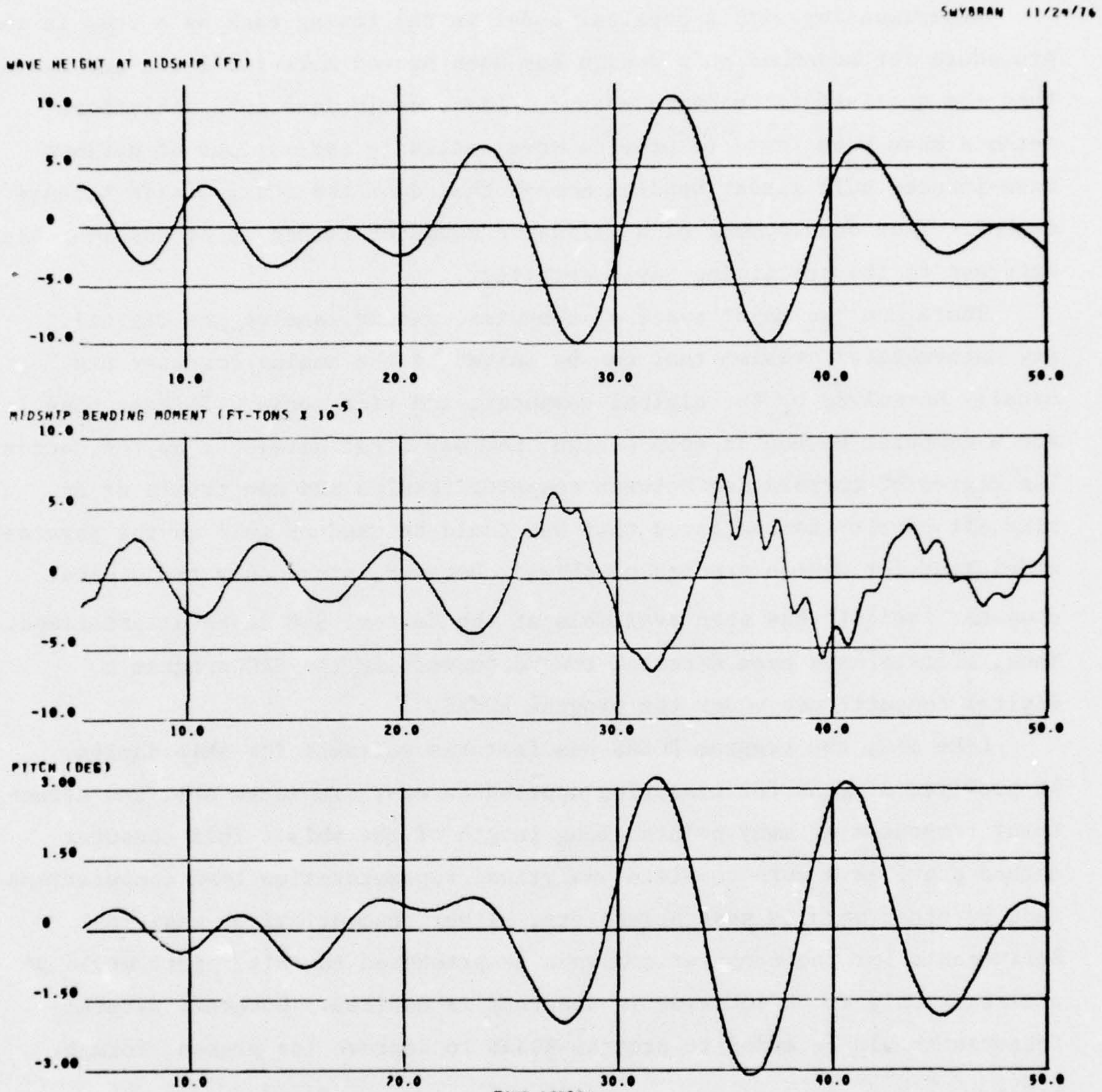


Figure 9 - Response to Bottom Slamming

SUMMARY AND CONCLUSIONS

Experimenting with a physical model in the towing tank as a step in the procedure for advanced ship design has been proved more realistic and reliable than the quasi-static balance method. Also, newly developed analytical methods have been found to provide more realistic estimations of maximum wave-induced hull girder bending moment than does the quasi-static balance method. Thus development of a validated computer method is of considerable interest to the practicing naval architect.

There are two major types of computer, namely, analog and digital. Any mathematical problem that can be solved by the analog computer can usually be solved by the digital computer, and vice versa. In searching for a computer method in ship design, SAC was first developed by the Center. The degree of correlation between computer results and sea trials of an aircraft carrier demonstrated that SAC could be used as well as the physical model test for design studies of ships. However, since only the digital computer facility has been available at the Center, SAC never materialized. Thus, efforts have been directed toward converting the SAC program to digital computer use under the program ROSAS.

Like SAC, the program ROSAS has features suitable for ship design. It provides a means for measuring applied forces, displacements, and structural responses at many points along length of the ship. This computer method provides a more complete analytical representation than computations made by other methods used heretofore, either theoretical or empirical. Refinements for the computer analysis as presented in this report would be necessary only if an increase in accuracy is desired. However, several features should be added to program ROSAS to improve its present format, i.e.,

1. An accurate method for determining both structural and hydrodynamic damping coefficients is needed. Unfortunately, one is still not available.

2. The present program is limited to the head sea condition. The program will be more flexible and like the actual operation of a ship in the seaway, if its capability can be increased from two- to three-dimensional form. This improvement can be accomplished by further development of the program.

3. Capability for investigating bottom slamming has been incorporated in the program. However, head-on wave impact, or any other form of wave impact, has not. This also needs further development.

This study has been confined to checking the feasibility and workability of the program ROSAS after its conversion from SAC. Comparisons made among results obtained from both SAC and ROSAS as well as from the sea trials, physical model, and other methods lead us to conclude confidently that the program ROSAS as it is will be an extremely useful tool for advanced ship research and design. Of course, there is room for improvement, as described previously.

The user's manual is provided in the appendixes. This includes the method for determining ship parameters and hydrodynamic forces and the complete computer program of an illustrated example used in the report.

ACKNOWLEDGMENTS

The authors express their thanks to Dr. J. C. Adamchak, Messrs. J. T. Birmingham, J. N. Andrews, and W. H. Buckley who reviewed the report and provided expert advice to the authors. The authors also gratefully acknowledge the helpful assistance by Mr. R. S. Sigman in preparing a portion of the users manual given in the appendixes. They also wish to take this opportunity to express thanks to Dr. W. W. Murray and Mr. A. B. Stavovy who originally suggested the work.

APPENDIX A - INPUT DATA FOR COMPUTER MODEL

At present, ROSAS is still in a preliminary stage of development. The original concept of this computer model was to have it supply a new ship design, specifying the lines and mass elastic parameters of a ship, which would be simulated on the computer for various environmental conditions. The hydrodynamic and buoyancy forces would then be computed automatically and supplied to various positions along the ship. This can easily be accomplished in time. Presently, the user has to either calculate all these values by longhand or write his own program.

Procedures for calculating parameters of ships are given in detail in Reference 8. The required parameters for computer input are:

1. Ship mass m
2. Bending stiffness EI
3. Mass moment of inertia I_{mz}
4. Shear stiffness KAG
5. Added mass m_V
6. Buoyancy force
7. Smith correction factor ρA_0
8. Structural damping coefficient C
9. Hydrodynamic damping $C(\omega)$. All calculations are based on 21 ship stations, with Stations 0, 1, 2, . . . , 20; Half-Stations 1/2, 1 1/2, . . . , 19 1/2; rudder post Station 0; bow Station 20. Procedures for calculating these parameters are given briefly as follows.

CALCULATION OF SHIP MASS

The total weight of each section — including hull structure, rudder, machinery, ballast, fuel, cargo, etc. — is assumed to be concentrated either at the center of that section or at the half-station point. Either weight distribution curves or a list of weights can be obtained from the shipyard or the ship design office. For a new ship, it may be obtained either by comparing the weight distribution of a sister ship or by estimation. All

[8] Bruck, H.A., "Procedure for Calculating Vibration Parameters of Surface Ships," NSRDC Report 2875 (Oct 1968).

PRECEDING PAGE BLANK-NOT FILMED

the weight added together shall be the displacement of a ship under any loading conditions.

The procedure for calculating ship mass distribution is as follows:

1. Obtain a weight distribution curve. This must be the total weight of the ship. If a list of weights is provided, construct a weight distribution curve.

2. Divide the weight distribution into 20 equal sections by drawing a perpendicular line at each station point; then, find the area under the curve for each section. This area is the weight of that section. The weight is assumed to be located at the center of each section, i.e., at the half-station point.

3. Make a table with three columns as indicated in Table 2. In the first column list the half-stations from 1/2 to 19 1/2, in the second list the weight just obtained from the curve, and in the third list the masses m. The mass is calculated by dividing each weight by the value of g, i.e., 32.2 ft/s^2, and then dividing the quotient by the length of each section to obtain mass per unit length of the section.

TABLE 2 - SHIP MASSES

Station	Weight W tons	Mass m $ton\text{-}s^2/ft^2$
1/2		
1 1/2		
.		
.		
.		
19 1/2		

Note: $m = W/g/\Delta X$

CALCULATION OF BENDING STIFFNESS

The value of Young's modulus E can easily be found from the materials handbook. The vertical area moment of inertia I for a ship section is calculated by summing the area moment of inertia of the deck, shell, double bottom, and continuous longitudinal members that comprise that section; it is determined about the horizontal neutral axis, which is parallel to the ship baseline through the centroid of that section, with

$$I = \Sigma\ (I_0 + ad^2) \tag{22}$$

where I_0 is the area moment of inertia of a structural member about its own neutral axis

a is the area of a structural member

d is the distance from the neutral axis of the structural member to the reference axis.

The procedure for calculating the vertical area moment of inertia I is as follows:

1. The decks and shell of a ship are usually constructed by using several plates of different thickness welded or riveted together. Find the areas of deck and shell plates, longitudinals, etc. List these areas in a table such as Table 3.

2. Use the baseline of the ship as the reference axis.

3. Measure and list the distance d of each item from its centroid to the baseline.

4. Multiply a by d and by d^2; calculate I_0, and list.

5. Calculate $d_0 = \Sigma\ ad/\Sigma\ a$, and $I = 2\ (\Sigma\ I_0 + \Sigma\ ad^2 - d_0^2\ \Sigma\ a)$; where d_0 is the distance of neutral axis of ship section from the baseline, and I is the total vertical area moment of inertia of ship section.

6. Repeat this procedure for all sections.

7. Plot all values of I against ship length and fair-in the curve. Locate the half-station points, read-off the I values at half-stations, and list.

8. The EI values are simply I multiplied by the constant E, which is 1.93×10^6 ton/ft^2 for steel.

TABLE 3 - DETERMINATION OF SECTIONAL VERTICAL AREA MOMENT OF INERTIA

Section No. ________

Item	a	d	d^2	ad	ad^2	I_0
	ft^2	ft	ft^2	ft^3	ft^4	ft^4
Deck Plate						
-						
-						
-						
Deck Long'l						
-						
-						
-						
Shell Plate						
-						
-						
-						
Shell Long'l						
-						
-						
-						
Double Bottom						
-						
-						
-						
Long'l						
-						
-						
-						
Etc.						
	Σ a			Σ ad	Σ ad^2	Σ I_0

$d_0 = \Sigma\, ad/\Sigma\, a$

$I = 2\ (\Sigma\, I_0 + \Sigma\, ad^2 - d_0^2\, \Sigma\, a)$

Not counted in the calculations are superstructures, discontinuous longitudinals, hatchways, deck plating between two hatches, and transverse members. However, welded intercostal members with no lightening holes should be included.

I_0 is usually omitted in calculating horizontal plates because I_0 values are very small, compared to ad^2 values.

CALCULATION OF MASS MOMENT OF INERTIA

The mass moment of inertia I_{mz} of a ΔX section of a ship about a rotating axis through its center of gravity parallel to z-axis (horizontal athwartship direction) consists of contributions from the ship mass and the added mass of fluid, i.e.,

$$\text{Total } I_{mz} = I_{mz}\big]_{hull} + I_{mz}\big]_{superstructure} + I_{mz}\big]_{added\ mass}$$

The added mass moment of inertia may be neglected.[8] For the hull, the actual mass in a length ΔX is assumed to be uniformly distributed and is bounded by the main deck and the shell. The total I_{mz} of a station is given by the equation

$$I_{mz} = m \cdot r^2 \tag{23}$$

where m is the mass per unit length of a ship section, and r is the radius of gyration given by the equation

$$r^2 = (d^2 + (\Delta X)^2)/12 \tag{24}$$

with d the depth of ship to main deck. If the ship deviates from standard, i.e., having its flight deck above the main deck or a long superstructure, detailed calculation to obtain I_{mz} is recommended.

CALCULATION OF SHIP SHEAR STIFFNESS

The ship hull is similar to a box girder with its uppermost continuous deck and bottom as flanges and its side shell as webs. In vertical vibration, the shear V is essentially carried by the side shell and any continuous longitudinal bulkheads. The shear stress is approximately uniform over the area of vertical plating A_v. This gives

$$KAG \cong A_v G \tag{25}$$

where K is the constant, depending on the shape of ship hull cross section

A is the cross sectional area of ship hull

G is the shear modulus of elasticity ($G = 7.72 \times 10^5$ ton/ft^2 for steel)

A_v is the cross sectional area of ship hull for vertical plating only.

CALCULATION OF ADDED MASS

A body moving with unsteady motion in an ideal fluid is subject to hydrodynamic pressure forces which are proportional to instantaneous acceleration. The resultant force acting on the body is directed opposite to that of the acceleration in the same manner as if an additional mass were attached to the system. It is therefore called added mass — sometimes virtual or hydrodynamic mass.

For an arbitrary cross sectional ship form, it is customary to calculate the added mass per foot of length by using the added mass values per foot of length of an infinite length of plate with width b. This value is amended by correction factors C_V and J, developed by various authors to allow for the finite length of an actual ship and the departure of its cross sectional shape from a rectangle. The formulas used for calculating added mass per unit length are as follows. For ship emergence (Figure 3)

$$m_V = m_0 + m_1 = \frac{2}{\pi} J\, C_V\, \rho b^2 \tag{26}$$

For ship immergence

$$m_V = m_0 + m_2 = \frac{\pi}{2} J\, C_V\, \rho b^2 \tag{27}$$

Here J is the longitudinal coefficient, depending on the ship length-to-beam ratio L/2b, and is given in Figure 10. The added mass coefficient is C_V, depending on the ship sectional area coefficient β and the beam-to-draft ratio 2b/d, and is given in Figure 11.[4] Equations (26) and (27) are the added mass for both port and starboard sides and do not need to be multiplied by two again.

As indicated in Figure 3, the water surfaces at the ship boundary are different during ship emergence and immergence. This causes the difference in added masses, mainly because the half-width used in the previous equations is b for the emergency and $\frac{\pi}{2}$ b for the immergence, due to rising of the water surface at the boundary.

Calculation of added mass used for program ROSAS is divided in two parts. For sections with no nonlinear terms considered, assume that they are wall sided and that only the added mass at the still waterline is calculated by using Equation (27). For sections with nonlinear terms considered, the added mass at each 1 ft of draft interval is calculated by Equation (26) for emergence and Equation (27) for immergence. The procedure used for calculating the added mass with nonlinear terms is listed as follows:

1. Obtain a body plan of the ship which shows the cross sectional profiles for 21 stations.

2. Obtain a Bonjean curve of areas for 21 stations. (If not available, make one.)

3. At each station, obtain half-breadth b and cross sectional area A for every draft d at 1-ft intervals; where b is from the body plan, and A is from the Bonjean curve.

4. Calculate $\beta = A/(2bd)$, L/2b, and 2b/d; obtain C_V and J from Figures 10 and 11.

5. Use Equations (26) and (27) to calculate the added mass $m_0 + m_1$ and $m_0 + m_2$ for ship emergence and immergence.

6. Calculate the added mass m_0 associated with a still waterline by Equation (27) as if the added mass has no nonlinear term.

7. Obtain the nonlinear added mass terms m_1 and m_2.

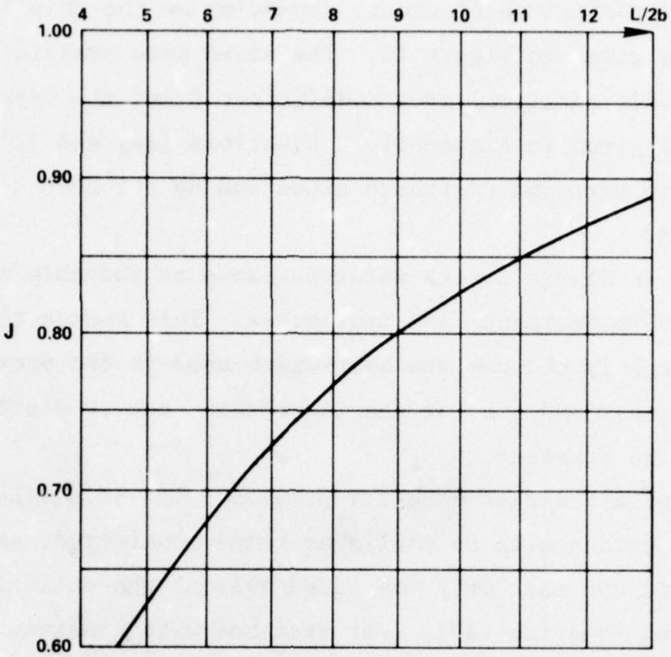


Figure 10 - Curve for Estimating Coefficient J Used in Added Mass Evaluation

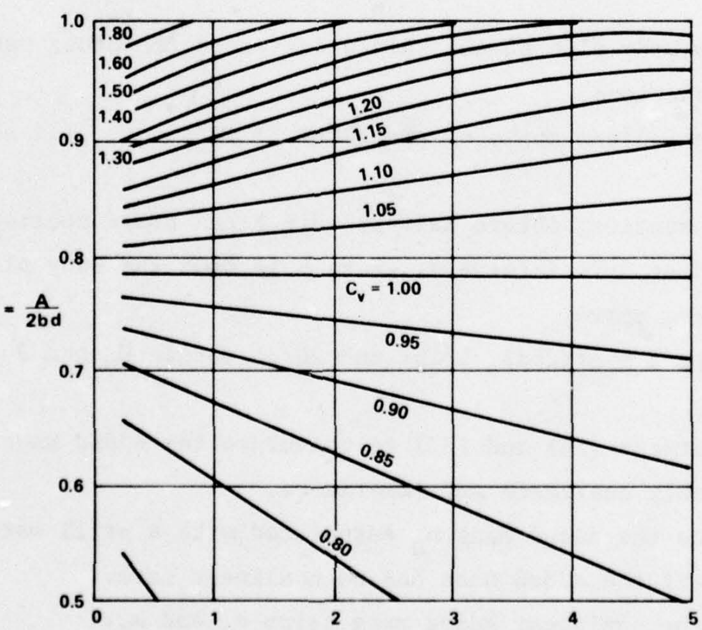


Figure 11 - Curves for Estimating Coefficient C_v Used in Added Mass Evaluation

8. Approximate m_1 and m_2 into a power series by

$$m_1 = a_1 Y_r + a_2 Y_r^2 + a_3 Y_r^3 \tag{28}$$

$$m_2 = c_1 Y_r + c_2 Y_r^2 + c_3 Y_r^3 \tag{29}$$

where a_1, a_2, a_3, c_1, c_2, and c_3 are arbitrary constants to be determined, e.g., as given in Reference 9.

9. The total added mass term for each section is given by Equation (16)

$$(m_V)_n = (m_0)_n + (a_1 Y_r)_n + (a_2 Y_r^2)_n + (a_3 Y_r^3)_n \tag{30}$$

for emersion, and

$$(m_V)_n = (m_0)_n + (c_1 Y_r)_n + (c_2 Y_r^2)_n + (c_3 Y_r^3)_n \tag{31}$$

for *immersion.*

CALCULATION OF BUOYANCY FORCE

The expression for the dynamic buoyancy force is given by Equation (20), in which

$$K_b = \rho g b_1 \Delta X/\Delta X \tag{32}$$

is the buoyancy spring, and b_1 is illustrated in Figure 2. The value of b_1 can be obtained either from the ship body plan or from the offset.

In calculating buoyancy force, the cross sectional area to the actual waterline is

$$A = A_0 - b_1 Y_r + \bar{A} \tag{33}$$

---

[9] Carnahan, B. et al., "Applied Numerical Methods," Chapter 1, John Wiley & Son Inc., New York (1969).

The buoyancy force per unit length of ship is then

$$\rho g A = \rho g \ (A_0 - b_1 \ Y_r + \bar{A}) \tag{34}$$

The nonlinear buoyancy force per unit length of ship becomes

$$\rho g \bar{A} = \rho g \ (A - A_0) + K_b \ Y_r \tag{35}$$

where A and A_0 are obtained from the Bonjean curve; thus, algebraic expressions can be determined in the form of

$$\rho g \bar{A} = b_2 \ Y_r^2 \ \rho g \tag{36}$$

for the nonlinear buoyancy force in Equation (20).

SMITH CORRECTION FACTOR

The Smith correction factor (SCF) ρA_0 is part of the dynamic buoyancy force given in Equation (20). The value of A_0 can be obtained as before by the Bonjean curve.

STRUCTURAL AND HYDRODYNAMIC DAMPING COEFFICIENTS

Evaluation procedures for determining damping of a ship seem uncertain at the present time. When a ship is oscillating on water, four types of damping factors are generated, namely,

1. Water Friction
2. Generation of a Pressure Wave
3. Generation of a Surface Wave
4. Structural Damping Force. The first three types generate the hydrodynamic damping force. Generation of surface waves and the structural damping force are the main sources of damping. McGoldrick (Reference 4, Table 8-2) mentioned from the analysis of many data about full-scale experimental work that damping in ship vibration appears to increase with frequency and the value of $(C + C(\omega))/m_s \omega$; where $(C + C(\omega))$ is the ship damping coefficient for the sum of hydrodynamic and structural damping, m_s is the ship

mass per unit length, and ω is the frequency in radians. He gave 0.034 as the mean value of $(C + C(\omega))/m_s\omega$ for all modes of ship vibration.

The rigid body motion of a ship mainly contains hydrodynamic damping in the low-frequency region. Determination of the hydrodynamic damping coefficient $C(\omega)$, which depends on the shape of the cross section of the ship, will be given later in the working example.

INPUT DATA FOR ESSEX

ESSEX has been used to test ROSAS because results obtained from ROSAS can be readily compared with those obtained from SAC,[1] the model test,[7] the sea trials,[2] and program SCORES.[7] The cellular divisions used in analyzing ESSEX are shown in Figure 4. Masses of the ship m are lumped at 20 equally spaced half-station points at intervals of 41 ft, and the bending rigidity EI is evaluated at the same points. These data are recorded in Table 4. The mass moment of inertia I_{mz} and the shear rigidity KAG are evaluated at 21 equally spaced stations and are recorded in Table 5.

Figure 4 also shows how the hydrodynamic forces are applied to the lumped model of the ship. The hydrodynamic forces are evaluated at nine stations — 3, 5, 7, 9, 11, 13, 15, 17, and 19. At four stations — 3, 15, 17, and 19 — nonlinear buoyancy and added mass effects are included. The hydrodynamic forces are equally distributed to either two or three half-station points in order to obtain a more uniform force distribution.

The linear hydrodynamic properties used in the analysis are listed in Table 6. Values apply to a ship in calm water at a draft of 28.5 ft.

The linear added mass m_0' for each station is equal to $\frac{\pi}{2} J C_V \rho b^2$ by Equation (27) and is listed in Table 7. The values for added mass are lumped, extending over the ship length from Station $(n - \frac{1}{2})$ to Station $(n + \frac{1}{2})$. Values for added mass listed in Table 6 come from Table 7. The value for m_{0_s}, for example, is

TABLE 4 - STRUCTURAL MASS AND BENDING RIGIDITY

Station	m	$EI \cdot 10^{-9}$
	ton-sec^2/ft^2	ton-ft^2
0.5	0.45467	5.05075
1.5	0.74401	7.76725
2.5	0.84735	12.31675
3.5	1.13281	18.30495
4.5	1.70245	27.04600
5.5	2.10287	36.95395
6.5	2.20362	44.24075
7.5	2.14550	49.67695
8.5	2.09512	55.16235
9.5	2.10158	58.75385
10.5	2.04374	59.00065
11.5	1.99308	50.67175
12.5	1.94270	37.47675
13.5	1.85357	33.20715
14.5	1.76574	34.4828
15.5	1.65465	28.64295
16.5	1.35756	19.27405
17.5	0.87060	13.65365
18.5	0.54122	9.82855
19.5	0.36426	7.4482

TABLE 5 - MASS MOMENT OF INERTIA AND SHEAR RIGIDITY

Station	$I_{mz} \cdot 10^{-3}$	$KAG \cdot 10^{-5}$
	$ton\text{-}sec^2$	ton
0	0.02686	13.3987
1	0.25408	16.3673
2	0.41905	24.8485
3	0.61563	35.3144
4	0.83089	47.7022
5	1.01557	60.5613
6	1.15716	71.4286
7	1.26788	77.6515
8	1.33742	78.3190
9	1.34124	75.0916
10	1.29996	71.8039
11	1.23465	69.6686
12	1.16907	70.2055
13	1.06159	75.2984
14	0.91029	81.5109
15	0.73649	73.2143
16	0.54697	40.0000
17	0.35450	24.5509
18	0.20379	18.5520
19	0.13025	14.3106
20	0.08431	12.6935

TABLE 6 - LINEAR ADDED MASS, BUOYANCY, AND SMITH CORRECTION FACTORS

Station	m_0	K_b	ρA_0
	ton-sec^2/ft^2	ton/ft^2	ton-sec^2/ft^2
3	1.2780	2.6057	0.9077
5	2.5988	3.1841	2.0060
7	3.5183	3.2951	2.6419
9	3.9610	3.3012	2.8519
11	3.7488	3.2732	2.7917
13	2.8329	3.0329	2.3768
15	1.5122	2.3220	1.5920
17	0.4963	1.2354	0.8216
19	0.1000	0.3402	0.3184

TABLE 7 - ADDED MASS AS A FUNCTION OF DESIGN WATERLINE OF 28.5 FEET

Station n	Added Mass m'_{0_n}	Station n	Added Mass m'_{0_n}	Station n	Added Mass m'_{0_n}
	ton-s^2/ft		ton-s^2/ft		ton-s^2/ft
0	0.23	7	146.06	14	89.87
1	18.12	8	157.78	15	60.58
2	37.34	9	164.27	16	36.97
3	59.43	10	163.27	17	18.19
4	84.20	11	155.58	18	7.96
5	107.44	12	140.39	19	3.48
6	127.03	13	117.14	20	0.71

Values for added mass are lumped extending over ship length from Station $(n - \frac{1}{2})$ to Station $(n + \frac{1}{2})$.

$$m_{0_5} = (\frac{1}{2} m'_{0_4} + m'_{0_5} + \frac{1}{2} m'_{0_6})$$

$$= (42.1 + 107.4 + 63.5)$$

$$= 213.1 \text{ ton-s}^2/\text{ft of 82 ft ship length}$$

$$= 2.5988 \text{ ton-s}^2/\text{ft}^2 \text{ of 1 ft ship length}$$

The linear buoyancy spring K_b is equal to $\rho g b_1$ given in Equation (20) and repeated in Equation (32). The values for the buoyancy spring listed in Table 4 are obtained in the same manner as for the added mass illustrated previously.

The Smith correction factor is equal to $\rho A_0 \, \Delta X/m_0$. The values of ρA_0 are listed in Table 4 and are obtained in the same manner as for the added mass. These three linear hydrodynamic properties of added mass, buoyancy spring, and Smith correction factor are obtained from Table 4 of Reference 1 which explains how these values were obtained.

The two nonlinear hydrodynamic forces, buoyancy force and added masses, are included for Station 3 at the stern and Stations 15, 17, 19 at the bow at the present time. This can be modified if so desired.

The nonlinear buoyancy forces used in the analysis are shown in Figure 12 as functions of emersion Y_r. These curves were derived from curves of the total buoyancy force shown in Figure 13 for a 1-ft section at Stations 3, 15, 17, and 19 by subtracting the linear spring rates recorded in Table 6. Algebraic expressions for these curves are given in Table 8. These were obtained by fitting the best mathematical curve to the data points for the nonlinear buoyancy force.

The nonlinear added masses used in the analysis are shown in Figure 14, and they were obtained in the same manner as for the nonlinear buoyancy force. Algebraic expressions for these curves are given in Table 9.

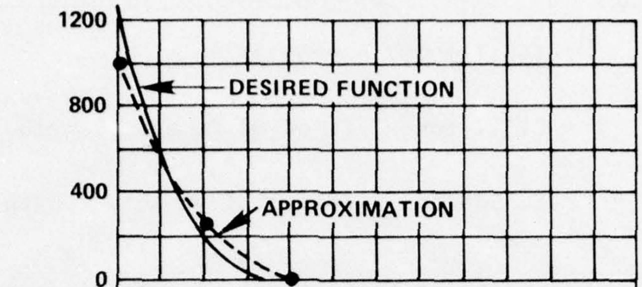


Figure 12a - Station 19

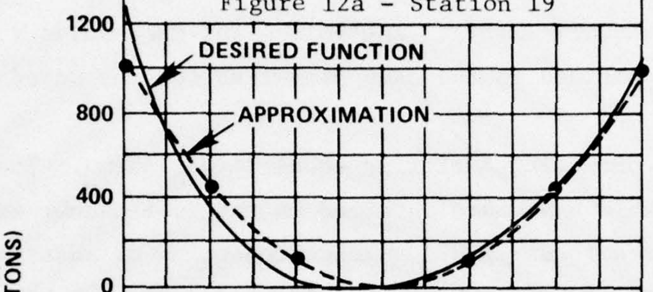


Figure 12b - Station 17

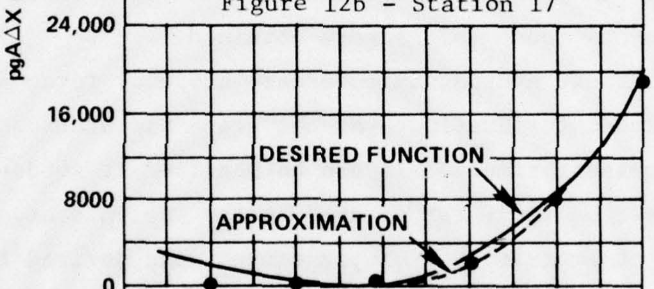


Figure 12c - Station 15

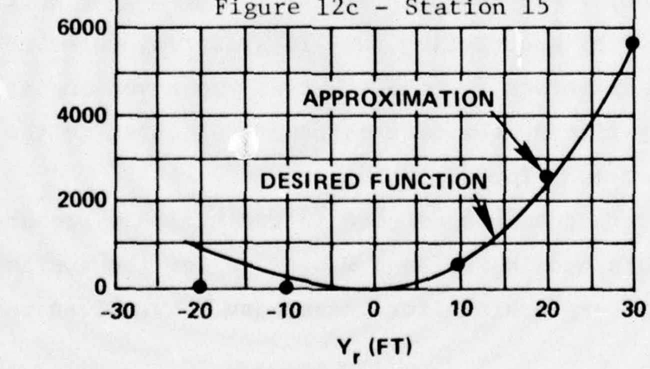


Figure 12d - Station 3

Figure 12 - Nonlinear Buoyancy Forces

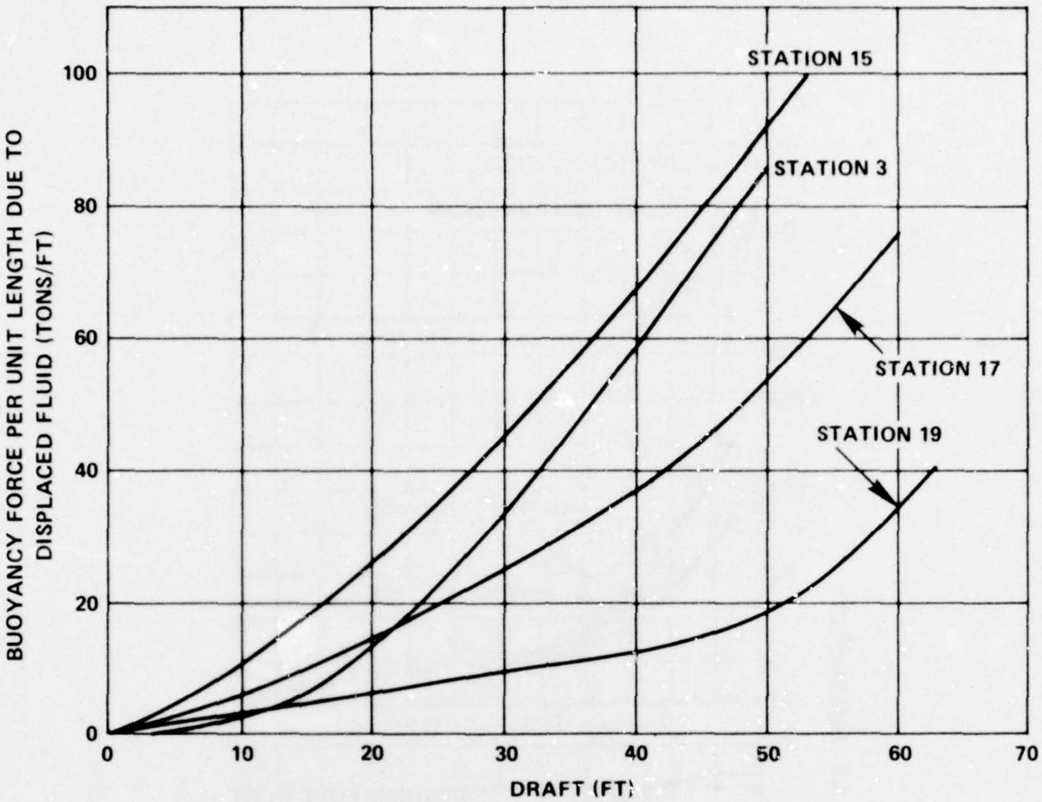


Figure 13 - Variation of Buoyancy Force with Draft

TABLE 8 - NONLINEAR BUOYANCY TERMS

Station	Buoyancy tons/ft	
19	$\rho g\bar{A} = 0.03024\ (Y_r + 10)^2$	$Y_r < -10$
	$= 0$	≥ -10
17	$\rho g\bar{A} = 0.01345\ Y_r^2$	
15	$\rho g\bar{A} = 0.02512\ Y_r^2$	$Y_r > 0$
	$= 0$	$Y_r \leq 0$
3	$\rho g\bar{A} = 0.05179\ Y_r^2$	$Y_r > 0$
	$= 0$	$Y_r \leq 0$

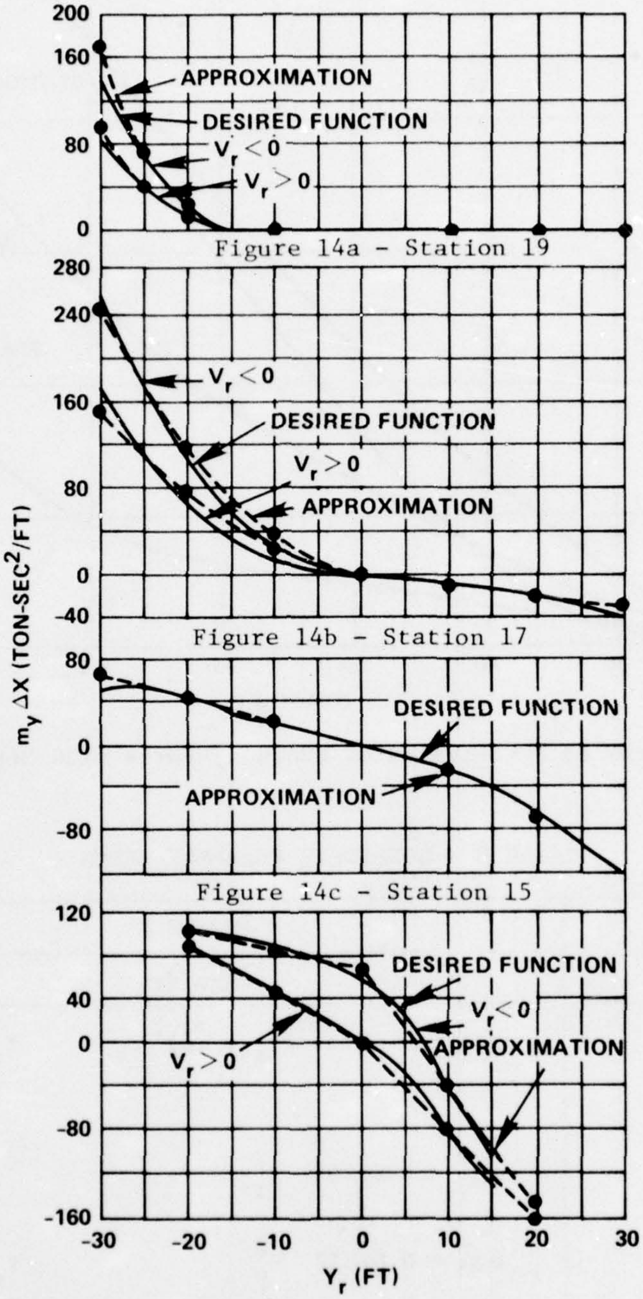


Figure 14d - Station 3

Figure 14 - Nonlinear Added Masses

TABLE 9 - NONLINEAR ADDED MASS TERMS

Station	Added Mass$\cdot 10^3$ ton-s^2/ft^2	
19	$\bar{m} = -0.1476\ (Y_r + 10)^3$	$Y_r < -10,\ V_r > 0$
	$= 0$	$> -10,\ > 0$
	$= -0.2658\ (Y_r + 10)^3$	$< -10,\ < 0$
	$= 0$	$> -10,\ < 0$
17	$\bar{m} = -13.17\ Y_r + 1.568\ Y_r^2$	$Y_r < 0,\ V_r > 0$
	$= -13.17\ Y_r$	$> 0,\ > 0$
	$= -13.17\ Y_r + 2.720\ Y_r^2$	$< 0,\ < 0$
	$= -13.17\ Y_r$	$> 0,\ < 0$
15	$\bar{m} = -26.13\ Y_r - 0.7805\ Y_r^2$	$Y_r > 0$
	$= -26.13\ Y_r$	< 0
3	$\bar{m} = -37.32\ Y_r$	$Y_r < 0,\ V_r > 0$
	$= -68.29\ Y_r$	$> 0,\ > 0$
	$= 560.2 - 14.74\ Y_r$	$< 0,\ < 0$
	$= 560.2 - 91.54\ Y_r$	$> 0,\ < 0$

Figure 15 shows the hydrodynamic damping coefficient for a 1-ft section. These curves were derived from References 10 and 11, and were used f^r the SAC given in Reference 1. Since damping force does not affect the solution given by the computer very much, the values of $C(\omega)$ used are considered sufficiently accurate for both SAC and program ROSAS.

Figure 15 can also be used to estimate $C(\omega)$ for other ships with nondimensional factors $\omega\sqrt{B/g}$ and $C(\omega)/(\Delta/\sqrt{gL^3})$. For ship vibration, the mean value of 0.034 for $(C + C(\omega))/m_s\omega$ may be used for all modes.

The velocity of wave propagation of the sea surface has been established

$$c(\text{fps}) = g/\omega \tag{37}$$

The wave velocity relative to the ship is equal to c+U.

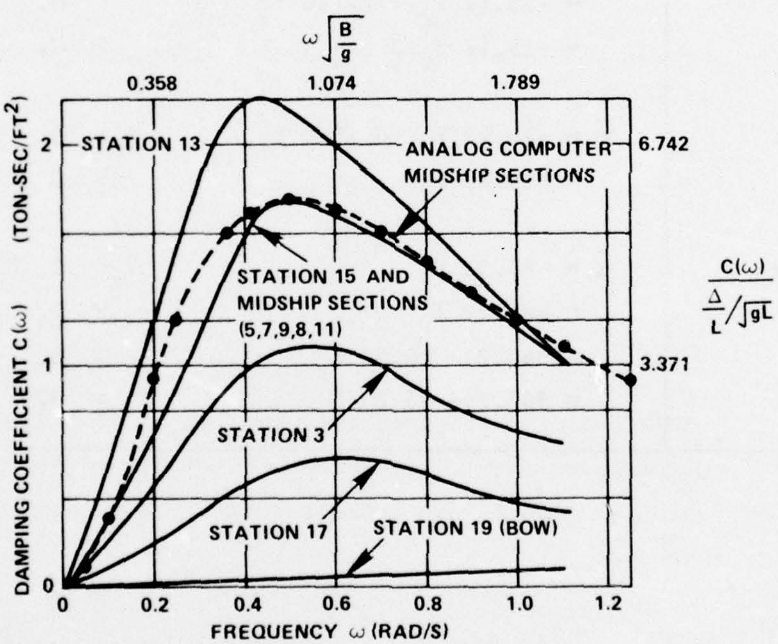


Figure 15 - Hydrodynamic Damping Coefficients

---

[10]Grim, O., "Berechnung der durch Schwingungen eines Schiffskörpers erzeugten hydrodynamischen Kräfte," Jahrbuch der Schiffbautechnischen Gesellschaft, Vol. 47 (1953).

[11]Golovato, P., "A Study of the Forces and Moments on a Heaving Surface Ship," David Taylor Model Basin Report 1074 (Sep 1957).

APPENDIX B - STRUCTURAL SEAWORTHINESS
DIGITAL COMPUTER PROGRAM ROSAS

The schematic of ROSAS has been shown in Figure 1. The program consists of the main program called PROGRAM SIMSHIP and many subroutines as shown in Figure 16. Descriptions of subroutines and functions are given in Table 10. The functioning of subroutines and alternate entries is given in Table 11, which shows names of subroutines to call and to be called. Table 12 gives the complete program ROSAS which includes PROGRAM SIMSHIP, the major subroutines SUBROUTINE HYD FRC, SUBROUTINE SLAM, SUBROUTINE SEA GEN, SUBROUTINE DAUX, SUBROUTINE KUTMER, and other subroutines. Also included in Table 12 are a list of input cards and a sample of selected output print. The flow chart for the program ROSAS is shown in Figure 17.

SIMSHIP

The main program is SIMSHIP. It determines the length of the time step to the next printing or plotting time. It calls the integration subroutine to integrate to this time and then prints data or calls subroutines that store data for plotting. At the end of the problem it calls subroutines to plot the data that have been previously stored. The flow chart for SIMSHIP is given by Figure 18.

KUTMER

SUBROUTINE KUTMER is used to integrate the system of ordinary differential equations that describe the response of the ship to hydrodynamic forces. This subroutine implements a Runge-Kutta method that incorporates automatic error control.

DAUX

SUBROUTINE DAUX is called by the integration routine to compute linear derivatives with respect to time of the ship vertical position, vertical velocity, angular velocity, bending moment, and shear force. These derivatives are computed at the various stations of the ship by evaluating the

expressions on the right-hand sides of Equations (15a) to (15d). The subroutine HYD FRC is called by DAUX to compute the hydrodynamic force P acting on the ship.

HYD FRC

SUBROUTINE HYD FRC computes the hydrodynamic force P, using Equations (1) to (4). The hydrodynamic forces are computed at odd numbered Stations 3 through 19. Then they are distributed to Half-Stations 1.5 through 19.5. The sum of the linear part of the added mass and the ship mass is returned to DAUX, where it is used in Equation (15a).

SUBROUTINES SEA GEN and SLAM are called by HYD FRC to compute a kinematic description of the surface of the sea and forces due to bottom slamming of the ship. Principal FORTRAN variables for subroutine HYD FRC are listed in Table 13, and the flow chart for subroutine SLAM is shown in Figure 19.

SEA GEN

SUBROUTINE SEA GEN computes the vertical height, velocity, and acceleration of the sea. These values are computed and returned for each of the nine stations at which the hydrodynamic forces are computed. Also the velocity of the waves is computed. For sinusoidal waves, vertical height, velocity, and acceleration of the waves is computed by

$$Y_w = H \sin \omega(t+X/c)$$

$$\dot{Y}_w = \omega H \cos \omega(t+X/c)$$

$$\ddot{Y}_w = -\omega^2 H \cos \omega(t+X/c)$$

where H is the prescribed wave height in feet

ω is the prescribed wave frequency in radius/second

X is the position of the station for which the values are being computed. The wave velocity (celerity) c equals g/ω, where g is the acceleration of gravity in feet/second^2. The shape of the discrete wave train is approximated by the sum of two sinusoidal waves.

INPUT CARDS

The input deck consists of a first-case input deck and either none or more subsequent-case input decks. Case decks are separated by 7/8/9 cards, and the last-case input deck is followed by two consecutive 7/8/9 cards.

The possible card types in the first-case input deck are listed in Table 14. The consequence of omitting an input card is explained in the last column of that table. Subsequent-case input decks consist of only card types on which one or more fields differ from the preceding case.

Figure 16 - Structural Seaworthiness Digital Computer Program ROSAS

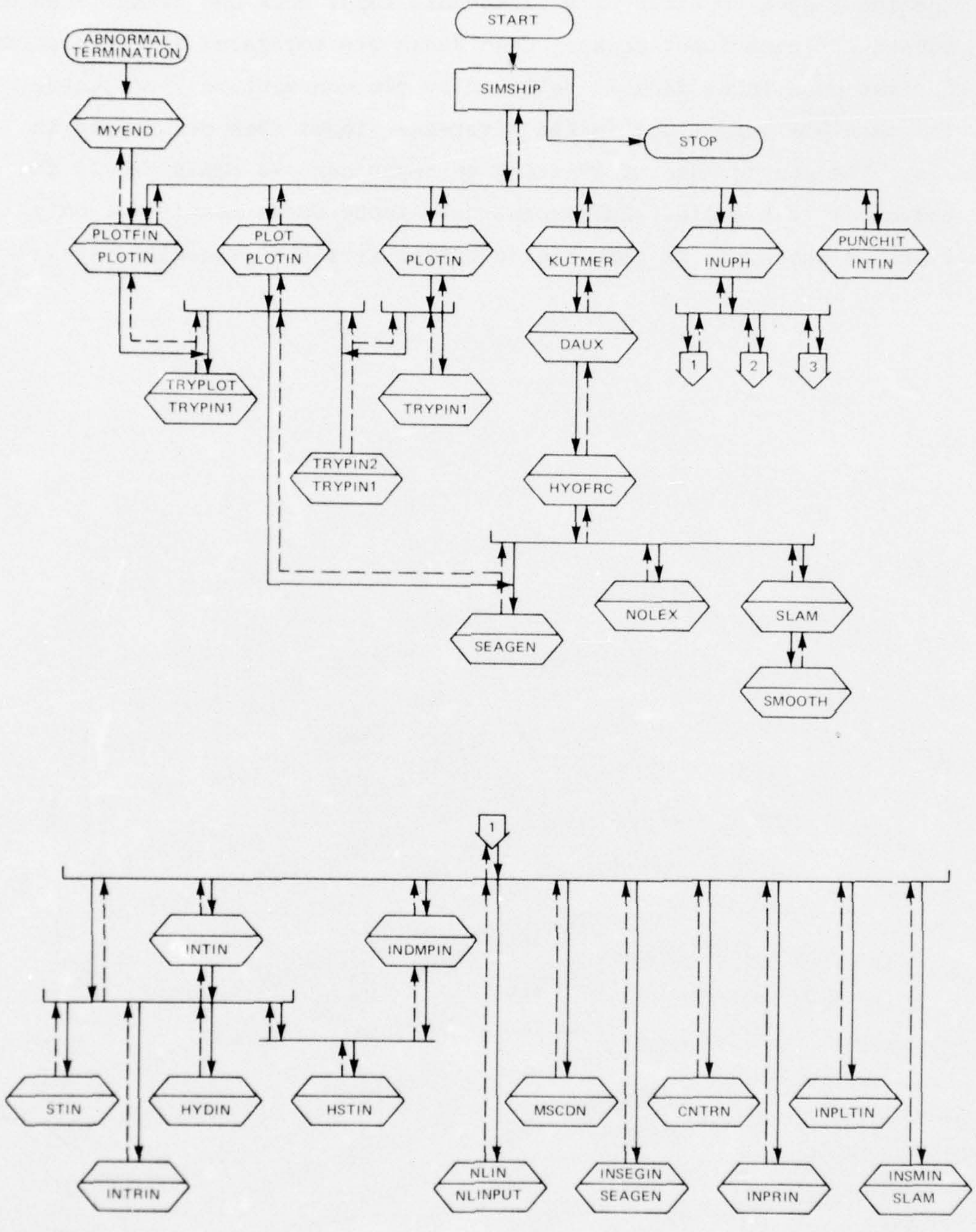
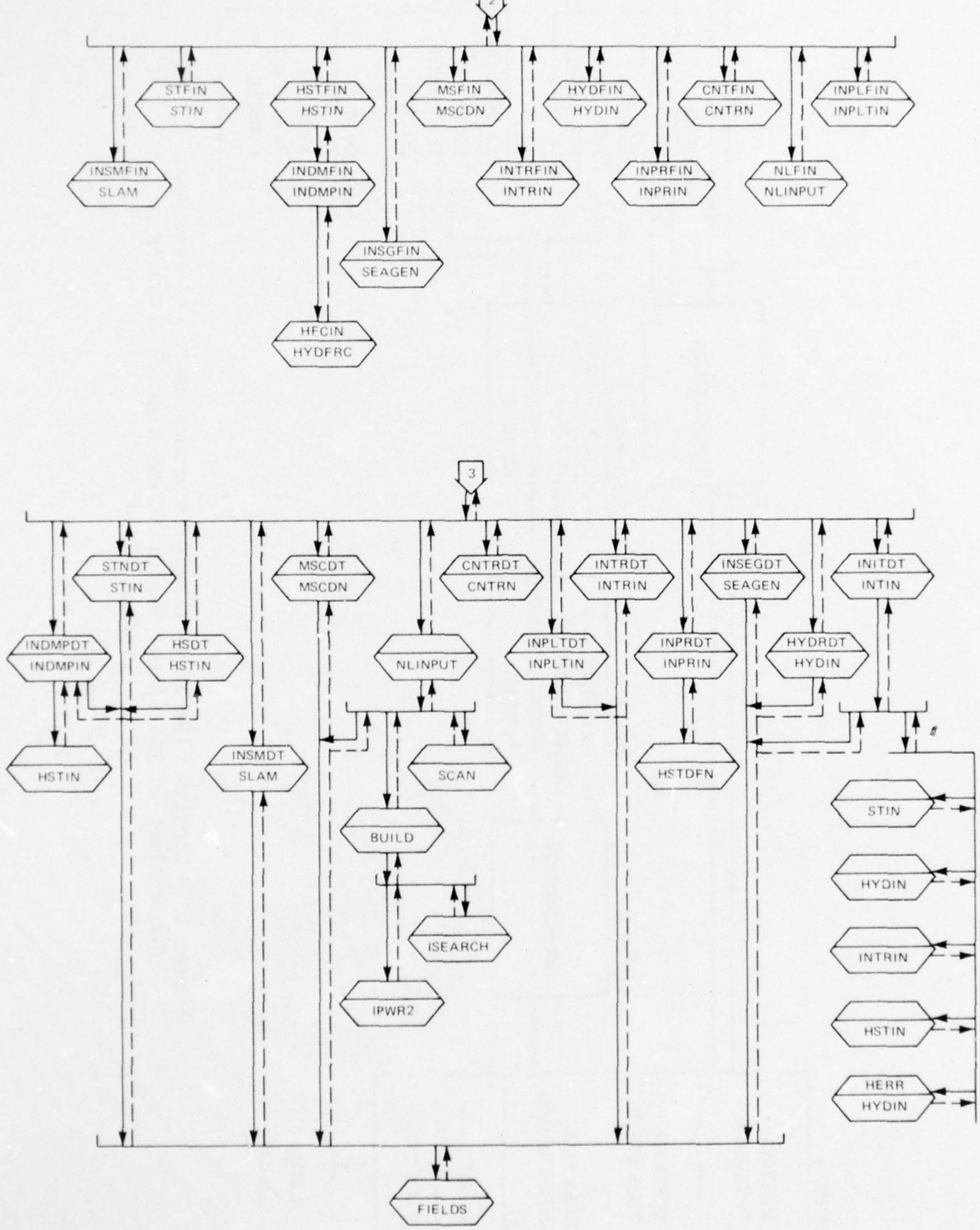


Figure 16 (Continued)



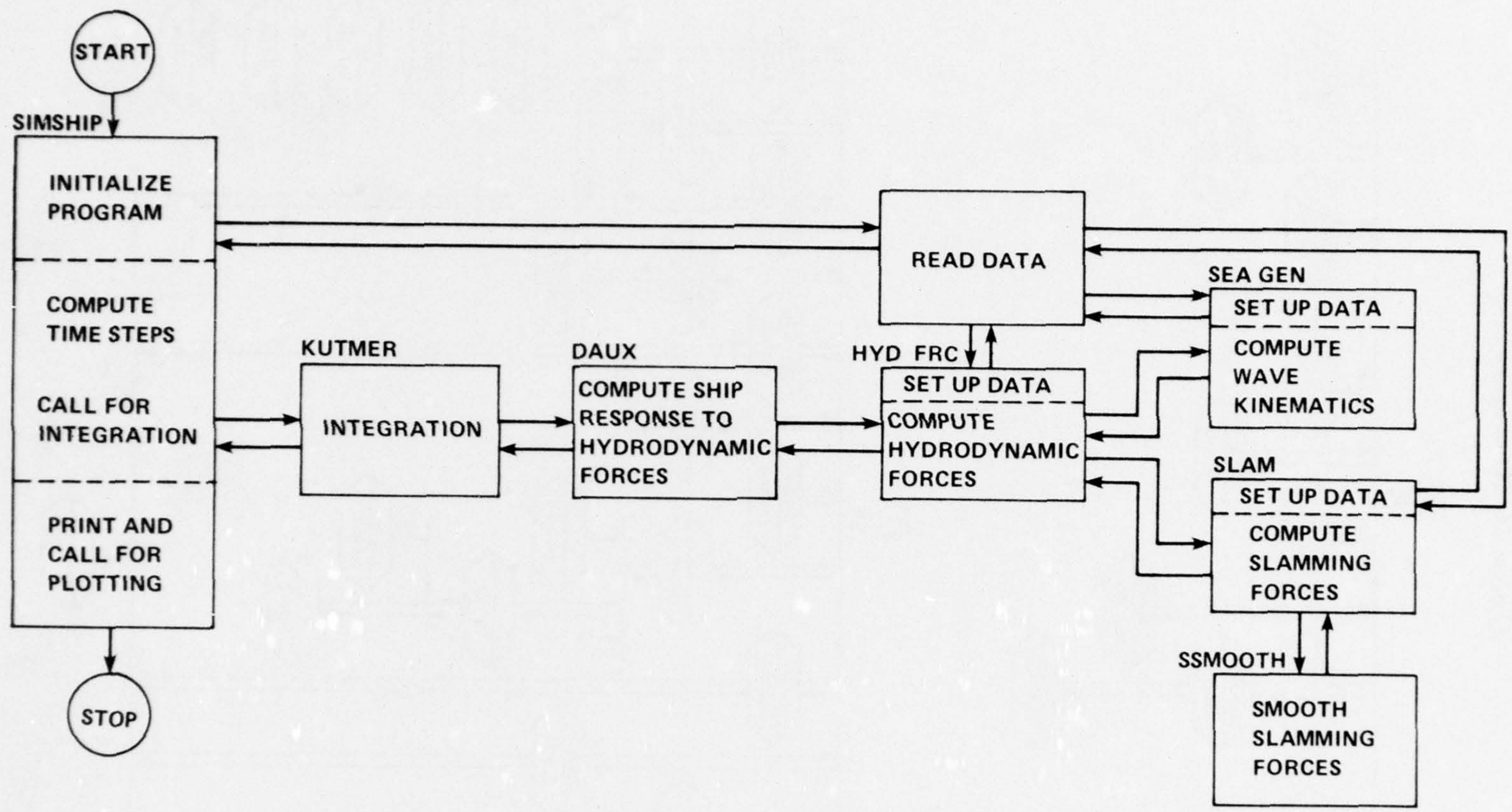


Figure 17 - Control Flow between Main Program SIMSHIP and Principal Subroutines of Structural Seaworthiness Digital Computer Program ROSAS

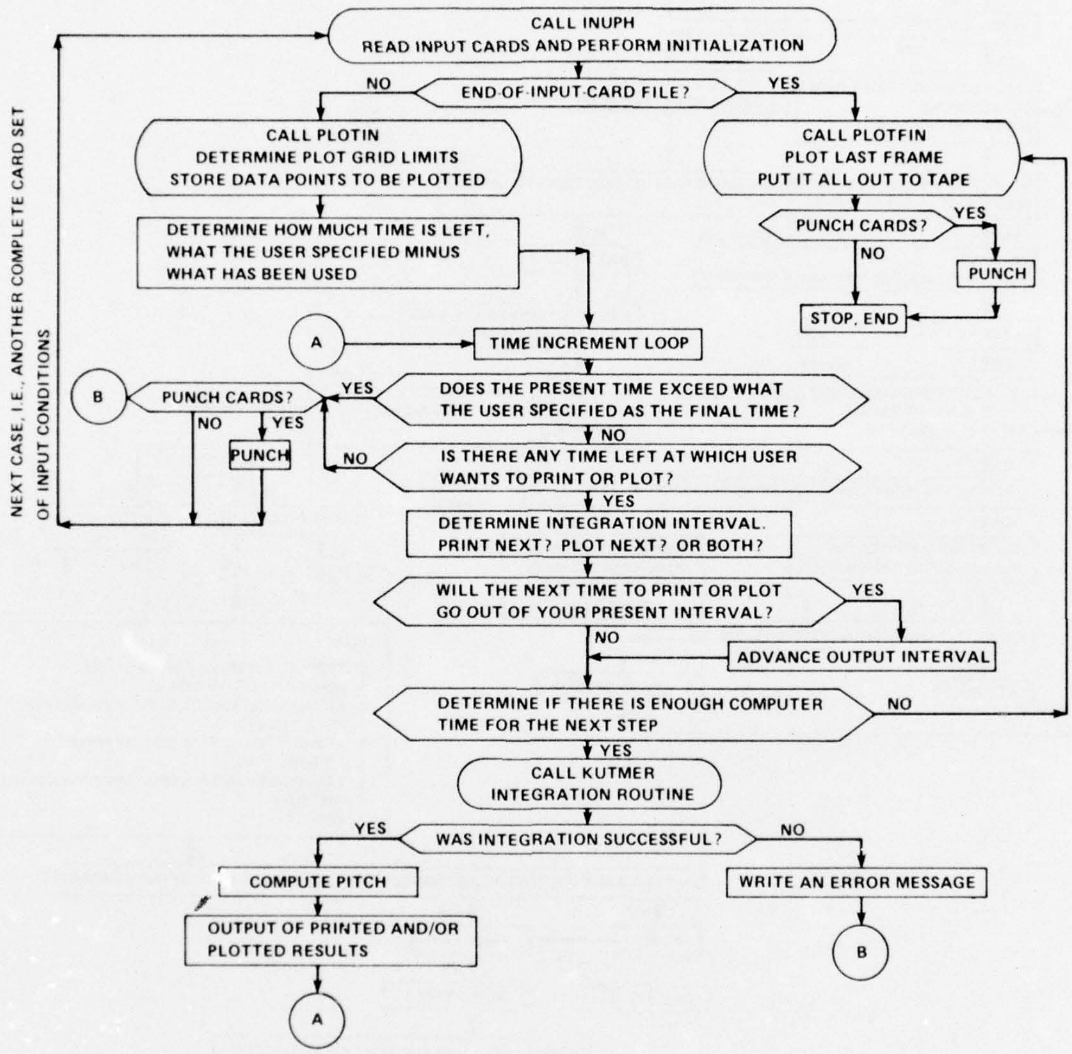


Figure 18 - Program SIMSHIP Flow Chart

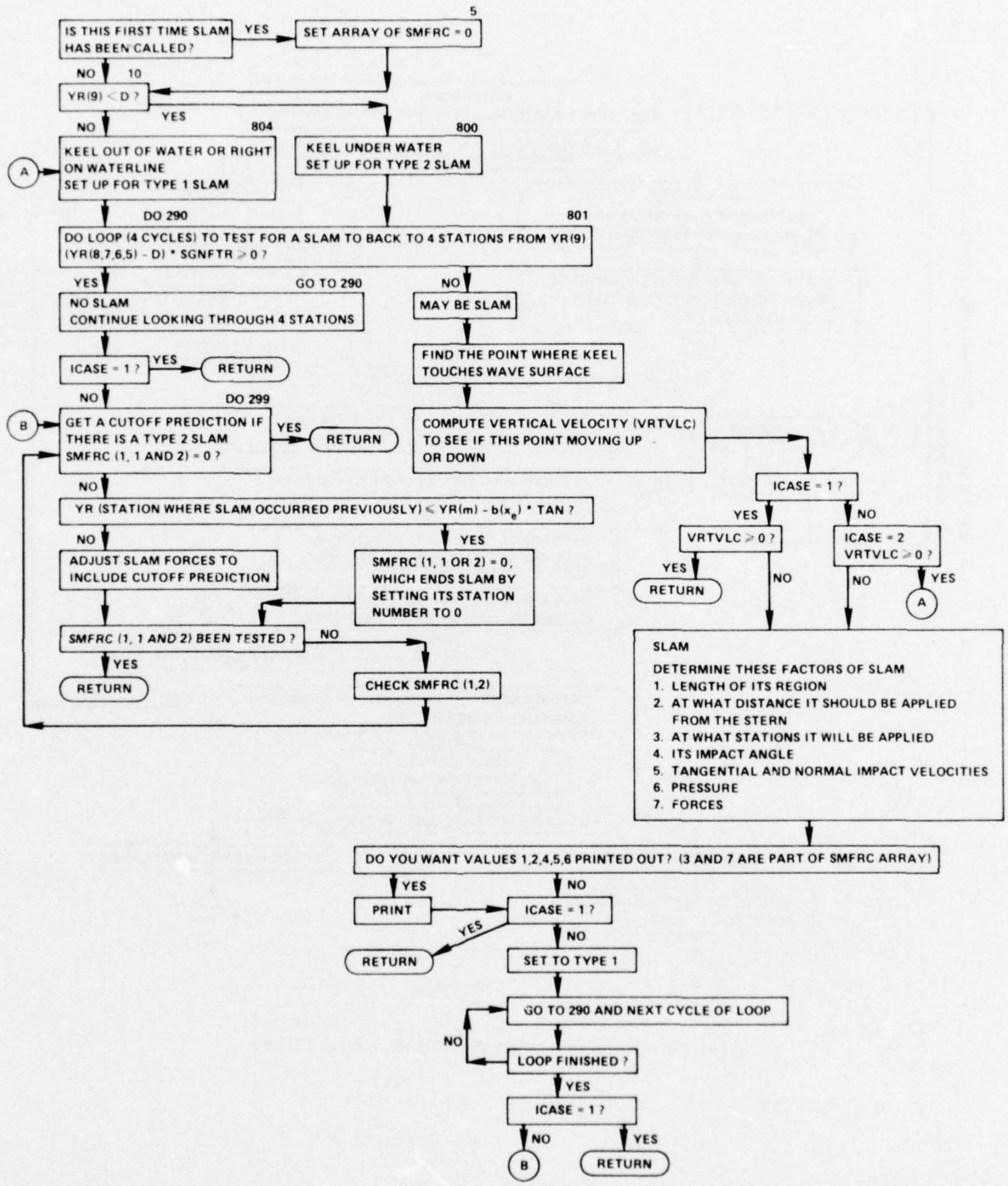


Figure 19 - Subroutine SLAM

TABLE 10 - GUIDE TO ROSAS SUBPROGRAMS

Name:	*Type**	*Description*
BUILD	S	Builds sorted list
CNTFIN	E(CNTRN)	Echoes CONTROL card data
CNTRDT	E(CNTRN)	Reads CONTROL cards
CNTRN	S	Sets CONTROL card defaults
DAUX	S	Computes derivative vector
FIELDS	S	Reads numeric data fields
HERR	E(HYDIN)	Prints diagnostic for illegally calculated hydroforce station
HFCIN	E(HYDFRC)	Initializes hydroforce calculations
HSDT	E(HSTIN)	Reads HALF-STTN cards
HSTDFN	S	Reads free-field bias-width definition
HSTFIN	E(HSTIN)	Echoes HALF-STTN card data
HSTIN	S	Sets HALF-STTN card defaults
HYDIN	S	Sets HYDRO card defaults
HYDFIN	E(HYDIN)	Echoes HYDRO card data
HYDFRC	S	Computes hydroforce
HYDRDT	E(HYDIN)	Reads HYDRO cards
INDMFIN	E(INDMPIN)	Echoes DAMPC card data
INDMPDT	E(INDMPIN)	Reads DAMPC cards
INDMPIN	S	Sets DAMPC defaults
INITDT	E(INTIN)	Reads INITIAL cards
IPWR2	F	Returns smallest power of two equally or exceeding argument
INPLFIN	E(INPLTIN)	Echoes PLOT and AXES card data
INPLTDT	E(INPLTIN)	Reads PLOT and AXES cards
INPLTIN	S	Sets PLOT and AXES card defaults
INPRDT	E(INPRIN)	Reads PRINT cards
INPRFIN	E(INPRIN)	Echoes PRINT card data
INPRIN	S	Sets PRINT card defaults
INSEGDT	E(SEAGEN)	Reads SEA GEN cards
INSEGIN	E(SEAGEN)	Sets SEA GEN card defaults

*E(nnnn) is alternate entry point of subprogram nnnn; F is function; S is subroutine.

TABLE 10 (Continued)

Name:	*Type**	*Description*
INSGFIN	E(SEAGEN)	Echoes SEA GEN card data; initializes sea generator calculations
INSMIN	E(SLAM)	Sets SLAM card defaults
INSMDT	E(SLAM)	Reads SLAM cards
INSMFIN	E(SLAM)	Echoes SLAM cards
INTIN	S	Initializes initial condition logic
INTRDT	E(INTRIN)	Reads INTGRTN cards
INTRIN	S	Sets INTGRTN card defaults
INTRFIN	E(INTRIN)	Echoes INTGRTN card data
INUPH	S	Executive routine for input data processing
ISEARCH	S	Performs binary search
KUTMER	S	Performs numerical integration
MSCDN	S	Sets SCTN and SPEED card defaults
MSCDT	E(MSCDN)	Reads SCTN and SPEED cards
MSFIN	E(MSCDN)	Echoes SCTN and SPEED card data
MYEND	S	Flushes buffers in the event of abnormal termination
NLIN	E(NLINPUT)	Sets NL_A/NL_B card defaults
NLINPUT	S	Reads NL_A/NL_B cards
NLFIN	E(NLINPUT)	Completes NL_A/NL_B card processing
NØLEX	F	Performs nonlinear expression evaluation
PLØT	E(PLØTIN)	Stores data to be plotted
PLØTFIN	E(PLOTIN)	Terminates plotting
PLØTIN	S	Initializes plotting
PUNCHIT	E(INTIN)	Punches final conditions
SCAN	S	Performs lexiographical scan
SEAGEN	S	Sea Generator
SLAM	S	Computes bottom slamming force
SSMOOTH	S	Smooths computed slamming force
STFIN	E(STIN)	Echoes STATION card data
STIN	S	Sets STATION card defaults
STNDT	E(STIN)	Reads STATION card data
TRYPIN1	S	Calculates vertical axes, plotting parameters
TRYPIN2	E(TRYPIN1)	Calculates horizontal axes, plotting parameters
TRYPLOT	E(TRYPIN1)	Plots one frame of data

*E(nnnn) is alternate entry point of subprogram nnnn; F is function; S is subroutine.

TABLE 11 - FUNCTIONING OF SUBROUTINES AND ALTERNATE ENTRIES

Name	Alternate Entries	Called By	Calls	Labeled Common Blocks
BUILD		NLINPUT	ISEARCH	
			IPWR2	
CNTRN	CNTFIN	INUPH		INPT2, INPT1
	CNTRDT			
DAUX		KUTMER	HYDFRC	HBLOK, TEST
FIELDS		INTIN		FBLOK
		HYDIN		
		SEAGEN		
		INTRIN		
		INPLTIN		
		NLINPUT		
		MSCDN		
		HSTIN		
		STIN		
		INDMPIN		
HSTDEN		INPRIN		
HSTIN	HSTFIN	INUPH	FIELDS	HBLOK, INPT2, FBLOK
	HSDT	INDMPIN		
		INTIN		

TABLE 11 (Continued)

Name	Alternate Entries	Called By	Calls	Labeled Common Blocks
HYDIN	HERR	INTIN	FIELDS	INPT2, HBLOK, FBLOK, HDBLOK
	HYDRDT	INUPH		
	HYDFIN			
HYDFRC	HFCIN	DAUX	SEAGEN	INPT1, NBLOCK, CBLOK, HBLOK, TEST, HDBLOK, PBLOK1
		INDMPIN	SLAM	
			NOLEX	
INDMPIN	INDMFIN	INUPH	HSTIN	HBLOK, FBLOK, INPT2
	INDMPDT		HYDFRC	
			FIELDS	
INPLTIN	INPLFIN	INUPH	FIELDS	INPT2, FBLOK, PBLOK1, PBLOK4, INPT1
	INPLTDT			
INPRIN	INPRDT	INUPH	HSTDFN	FBLOK, HBLOK, INPT1, INPT2
	INPRFIN			
INTIN	INITDT	INUPH	STIN	HBLOK, FBLOK, INPT1, INPT2
	PUNCHIT		INTRIN	
			HYDIN	
			HSTIN	
			FIELDS	
INTRIN	INTRDT	INUPH	FIELDS	FBLOK, INPT1, INPT2
	INTRFIN	INTIN		

TABLE 11 (Continued)

Name	Alternate Entries	Called By	Calls	Labeled Common Blocks
INUPH		SIMSHIP	CNTRN	INPT1, INPT2
			HYDIN	
			HSTIN	
			INDMPIN	
			INPLTIN	
			INPRIN	
			SEAGEN	
			INTRIN	
			MSCDN	
			NLINPUT	
			INTIN	
			STIN	
ISEARCH		BUILD		
KUTMER		SIMSHIP	DAUX	
MSCDN	MSCDT	INUPH	FIELDS	FBLOK, HBLOK, INPT1, INPT2
	MSFIN			
MYEND		ABNORMAI TERMINATION	PLOTIN	
NLINPUT	NLIN	INUPH	FIELDS	FBLOK, INPT2, CBLOK
	NLFIN		BUILD	
			SCAN	
NOLEX		HYDFRC		NBLOCK

TABLE 11 (Continued)

Name	Alternate Entries	Called By	Calls	Labeled Common Blocks
PLOTIN	PLOT	SIMSHIP	TRYPIN1	PBLOK1, CBLOK, PRNT1, PBLOK3
	PLOTFIN	MYEND	SEAGEN	
SCAN		NLINPUT		
SEAGEN	INSEGDT	INUPH	FIELDS	FBLOK, INPT2, HBLOK
	INSEGIN	PLOTIN		
	INSGFIN	HYDFRC		
SLAM	INSMIN	INUPH		
	INSMDT	HYDFRC	SSMOOTH	HBLOK, TEST
	INSMFIN			
SSMOOTH		SLAM		
STIN	STFIN	INUPH	FIELDS	HBLOK, INPT2, FBLOK
	STNDT	INTIN		
TRYPIN1	TRYPIN2	PLOTIN		PBLOK3, PBLOK4
	TRYPLOT			

TABLE 12 - STRUCTURAL SEAWORTHINESS DIGITAL COMPUTER PROGRAM ROSAS

```
      PROGRAM SIMSHIP(INPUT=128,OUTPUT=256,PUNCH=128,TAPE5=INPUT,TAPE8,   SMSHP
     1 TAPE6=OUTPUT,TAPE7=PUNCH,FILMPL=128,TAPE48=FILMPL,DEBUG=OUTPUT)    SMSHP
C$    DEBUG                                                               SMSHP
C$    ARRAYS                                                              SMSHP
      DIMENSION TP(2), BB(2),TPP(2),L(2)                                  SMSHP
      DIMENSION YH(9)                                                     SMSHP
      COMMON /HBLOK/ KAG(21),RINT(21),EI(20),SHPMS(20),DAMPC(20),         SMSHP
     1 DXI,NEVAL,SPEED,BUOY SPG(9),SCF(9),ADDMS(9),VRTMS(20),IBEG,IEND    SMSHP
      COMMON/INPT1/IEOF,B(11,2),TSTRT,HMNI,TF,W(10,2),NW(2),EPS(91) ,     CD5
     1ABSFAR(91),XLNGTH ,TCPU,CONTROL(4)                                  CD5
      COMMON /TEST/ IND                                                   SMSHP
      DIMENSION       YDOT(20),GDOT(21),BDM(20),SF(21),TEMP(21)           SMSHP
      EXTERNAL MYEND                                                      SMSHP
      EQUIVALENCE                                                         CD3
     1 (YS,YH),(YS(10),YDOT),(YS(30),GDOT),(YS(51),BDM),(YS(71),SF)       CD3
      COMMON /PRNT1/T,PITCH,TLEFT                                         SMSHP
      EQUIVALENCE (YS,YH)                                                 SMSHP
      COMMON YS(91)                                                       SMSHP
      DATA IPG,YES / 0,3HYES /                                            SMSHP
*                                                                         SMSHP
C**INITIALIZATION                                                         SMSHP
*                                                                         SMSHP
 5    CALL SECOND(TC1)                                                    SMSHP
      CALL INUPH                                                          SMSHP
      IF (IEOF .EQ. 1) GO TO 1050                                         SMSHP
      CALL PLOTIN                                                         SMSHP
      IND = 0                                                             SMSHP
      CALL SECOND(TC2)                                                    SMSHP
      TLEFT=.9*TCPU-TC2+TC1                                               SMSHP
      DO 10 K=1,2                                                         SMSHP
      TP(K)=B(1,K)                                                        SMSHP
      BB(K)=(1.+SIGN(.0001,B(2,K)))*B(2,K)                                SMSHP
      TPP(K)=(1.-SIGN(.0001,TP(K)))*TP(K)                                 SMSHP
 10   L(K)=1                                                              SMSHP
      T=TSTRT                                                             SMSHP
      NEVAL=0                                                             SMSHP
      ISTEPS=0                                                            SMSHP
*                                                                         SMSHP
C**TIME-INCREMENT LOOP                                                    SMSHP
*                                                                         SMSHP
 100  IF (T.GT.TF) GO TO 900                                              SMSHP
      IF (TP(1) .GE. 1.E20 .AND. TP(2) .GE. 1.E20) GO TO 900              SMSHP
C**DETERMINE INTEGRATION INTERVAL                                         SMSHP
*                                                                         SMSHP
*   WHICH IS NEXT?  PRINT(I=2), PLOT(I=1) OR BOTH(FALL THROUGH)           SMSHP
      DO 130 I=1,2                                                        SMSHP
      J=3-I                                                               SMSHP
      IF (TP(I) .LE. TPP(J)) GO TO 150                                    SMSHP
 130  CONTINUE                                                            SMSHP
      TT=(TP(1)+TP(2))*.5                                                 SMSHP
      K1=1                                                                SMSHP
      K2=2                                                                SMSHP
      GO TO 160                                                           SMSHP
 150  TT=TP(I)                                                            SMSHP
      K1=K2=I                                                             SMSHP
 160  H=TT-T                                                              SMSHP
C**DETERMINE NEXT PLOT/PRINT TIME                                         SMSHP
      JIND=0                                                              SMSHP
      DO 200 K=K1,K2                                                      SMSHP
      LL=L(K)                                                             SMSHP
      TTP=TP(K)+W(LL,K)                                                   SMSHP
*  WILL THE NEXT TIME GO OUT OF PRESENT INTERVAL, IF SO,                  SMSHP
*  ADVANCE INTERVAL                                                       SMSHP
      IF (TTP .GT. BB(K)) GO TO 170                                       SMSHP
      TP(K)=TTP                                                           SMSHP
      GO TO 195                                                           SMSHP
*  DOES THE NO. OF REQUESTED INTERVALS EXCEED THE NO. GIVEN, IF SO,       SMSHP
*  SET "INFINITE" STARTING TIME FOR THE NEXT OUTPUT.                      SMSHP
 170  IF (LL .LT. NW(K)) GO TO 180                                        SMSHP
```

```
      TP(K)=1.E20
      GO TO 195
  180 LL=L(K)=L(K)+1
      TP(K)=B(LL,K)+H(LL,K)
      BB(K)=(1.+SIGN(.0001,B(LL+1,K)))*B(LL+1,K)
  195  TPP(K)=(1.-SIGN(.0001,TP(K)))*TP(K)
      JIND=JIND+K
  200 CONTINUE
C**DETERMINE IF ENOUGH CPU TIME LEFT
      IF (ISTEPS .EQ. 3) GO TO 210
      ISTEPS=ISTEPS+1
      GO TO (220,212,230) ISTEPS
  212 TCOMP2=0
      GO TO 230
  210 IF(H1 .NE. H2)          GO TO 215
      H1 = 0
      TCOMP1 = 0.
  215 FRACT = (H - H1)/(H2 - H1)
      TCOMP = TCOMP1 + FRACT*(TCOMP2 - TCOMP1)
      IF (TLEFT-TCOMP.GT.0.) GO TO 230
      WRITE(6,850) H1,H2,H,TCOMP1,TCOMP2,TCOMP,TLEFT
  850 FORMAT(25H COMPUTED TIME LIMIT         /
     1        22H LAST TWO TIME STEPS    ,1P2E10.2,5X,15HNEXT TIME STEP
     2,1PE10.2 / 22H LAST TWO COMP TIMES     ,2F10.3,5X,
     3 15HNEXT COMP TIME    ,F10.3 / 47X,15HCOMP TIME LEFT    ,F10.3)
      IF(CONTROL(2) .EQ. YES)           CALL PUNCHIT
      GO TO 1050
C**START SIMULATION WITH 2 BABY STEPS
  220 TIME=0
      HH=1.E-05*HMNI
      FIRST=0.
      DO 225 KL=1,2
      HMIN=HMNI
      CALL SECOND (TIME1)
      CALL KUTMER(91,T,HH,YS,EPS,ABSERR,HMIN,FIRST)
      CALL SECOND(TIME2)
      TIME=TIME+TIME2-TIME1
      IF (FIRST .EQ. 2.) GO TO 875
  225 CONTINUE
      H2=HH
      H=H-HH-HH
      IF (H .LE. 0) GO TO 300
C**INTEGRATE EQUATIONS
  230 IF (H .EQ. H2) GO TO 240
      HMIN=H
      FIRST=.5
  235 IF(HMIN/HMNI .LT. 1.0001)        GO TO 240
      HMIN=HMIN/2
      GO TO 235
  240 CONTINUE
      IF (H .LE. 0) GO TO 100
      CALL SECOND(TIME1)
      CALL KUTMER(91,T,H,YS,EPS,ABSERR,HMIN,FIRST)
      CALL SECOND(TIME2)
      IF(ABS(1.-H2/H) .LT. .01)       GO TO 250
      H1 = H2
      TCOMP1 = TCOMP2
  250 TCOMP2 = TIME2 - TIME1
      TCOMP1=TCOMP2
      TLEFT=TLEFT-TCOMP2
      TIME=TIME+TCOMP2
      H2=H
      IF (FIRST .EQ. 2.) GO TO 875
*
C**COMPUTE THE PITCH
*
  300 PITCH=57.296*ATAN((YH(9)-YH(1))/XLNGTH)
*
C** OUTPUT RESULTS AND THEN JUMP BACK FOR ANOTHER TIME INCREMENT
*
```

Table 12 (Continued)

```
      IF (JIND .EQ. 2) GO TO 350
      CALL PLOT
      IF (JIND .EQ. 1) GO TO 100
  350 INDSAV = IND
      TX=(1.+1.E-10)*T
      IND = 1
      CALL HYDFRC (TX,YH,YDOT,TEMP)
      IND = INDSAV
      IPG = IPG + 1
      WRITE(6,49) T,TIME,HMIN,NEVAL
   48 FORMAT(1H1)
   49 FORMAT(1H-/// 5H TIME,F14.5,5H SECS,F17.3,14H SECS CPU TIME /
     1 10H TIME STEP ,F9.5,5H SECS,I17,23H DERIVATIVE EVALUATIONS)
      WRITE (6,50)
      WRITE (6,55) (YH(I),   I =  1, 9)
      WRITE (6,51)
      WRITE (6,55) (YDOT(I), I =  1,20)
      WRITE (6,56) PITCH
      WRITE (6,52)
      WRITE (6,55) (GDOT(I), I =  1,21)
      WRITE (6,53)
      WRITE (6,55) (BDM(I) , I =  1,20)
      WRITE (6,54)
      WRITE (6,55) (SF(I)  , I =  1,21)
   50 FORMAT (20H0VERTICAL POSITION   ,1P10E10.2)
   51 FORMAT (20H0VERTICAL VELOCITY   ,1P10E10.2)
   52 FORMAT (20H0ANGULAR VELOCITY    ,1P10E10.2)
   53 FORMAT (20H0BENDING MOMENT      ,1P10E10.2)
   54 FORMAT (20H0SHEAR FORCE         ,1P10E10.2)
   55 FORMAT ( 1H                     ,1P10E12.4)
   56 FORMAT (20H0PITCH               ,1P10E10.3)
      GO TO 100
*
C**COME HERE FOR CASE TERMINATION
*
 875  WRITE(6,880)
 880  FORMAT(1X,*STOPPED WITH FIRST  = 2.*)
 900  IF (CONTROL(2) .EQ. YES) CALL PUNCHIT
      GO TO 5
*
C** JOB TERMINATION
*
 1050 CALL PLOTFIN
      STOP
      END
```

Table 12 (Continued)

```
      SUBROUTINE HYDFRC (T,YH,YHDOT,HF)
C$    DEBUG
C$    STORES(HFTEMP,VMTEMP,KCOND,IAD)
      REAL NOLEX
      REAL KAG
      DIMENSION KAG(21),RINT(21),EI(20),SHP MS(20),DAMPC(20)
     1          ,SCF(9),BUOY SPG(9),ADD MS(9),            VRT MS(20)
     2          ,YH(9),YHDOT(20),YR(9),YRDOT(9),YWDD(9),VR(9),HF(21)
     3          ,YW(9),YWDOT(9)
     4          ,AM(2),BS(2),SC(2),HD(2)
     5          ,SMFRC(6,2), SMPTS(12)
      DIMENSION CS1(10)
C-----------------------------------------------------------------------
      DIMENSION NLINV(9), ZBY4DXI(9),HFTEMP(9),IBNL(5,9),ITYPE(18)
      DIMENSION YCRIT(18), VCRIT(18)
      DIMENSION IBSTORE(350),BSTORE(350),VMTEMP(9), IMNL(5,9)
      DIMENSION IMSTORE(350,2),AMSTORE(350,2),NAPPLY(20),NL(9)
      EQUIVALENCE (IBSTORE,BSTORE),(IMSTORE,AMSTORE)
      COMMON/NBLOCK/YDELT
      COMMON/CBLOK/NCALC,NHSTNS,NGROUP,NL,NLINV,IMNL,IBNL,ITYPE,YCRIT
     2,VCRIT,IBSTORE,IMSTORE,KWORDS(2),NP(2),IFIRST,NAPPLY
      COMMON /HBLOK/ KAG,RINT,EI,SHPMS,DAMPC,DXI,NEVAL,SPEED,
     1 BUOY SPG,SCF,ADDMS,VRTMS,IBEG,IEND
      COMMON /HDBLOK/ HYDDAMP(9)
      COMMON/PBLOK1/NR,TTT(20),R(20),MSBSTN,NAME,CODE,PHONE
      COMMON/INPT1/IEOF,B(11,2),TSTRT,HMNI,TF,W(10,2),NW(2),EPS(91) ,
     1ABSERR(91),XLNGTH ,TCPU,CONTROL(4)
      COMMON /TEST/ IND
      DATA TYR,TYRDOT,TYWDD,TVR,TAM,THF1,THF2,THF3,THF4,THF5,TVM,THF6
     1     ,THF7,TWH,TWV,SLM/
     1      9H REL DISP,8H REL VEL,9H WAVE ACC,9H RL WV VL,9H ADD MASS
     2     ,9H BUOY SPG,9H N-L B SP,9H N-L A MS,8H SPC DRV,4H SCF
     3,7H VRT MS,8H HYD FRC,8H HYD DMP,7H WV HGT,7H WV VEL,8H SLM FRC/
      DATA SMOTH/7H SMOOTH/
      DATA PI,LABEL / 3.14159265,0 /,AM,BS,SC,HD/1H ,10HADDED MASS,
     1      10H        BU , 10HOY. SPRING ,1H ,10HSMITH C.F.
     2     ,10H        HYD,10HRO DAMPING/,YES      /  3HYES     /,TWO PI
     3     / 6.283185307  /
      DATA CS1 / .000,.174,.342,.500,.643,.766,.866,.940,.985,1.00 /
      IF (NEVAL .EQ. IBEG)   IND = 1
      IF (NEVAL .EQ. IEND)   IND = 0
      IF(T .EQ. OLD T)                   RETURN
      OLD T = T
  100 CONTINUE
      IM=0
      IB=0
      CALL SEAGEN(YW,YWDOT,YWDD,CEL,T)
   71 CONTINUE
*
*  COMPUTE THE SHIPS VERTICAL DISPLACEMENT AND VELOCITY RELATIVE
*  TO THE SEA
      DO 110 I=1,NCALC
      YR(I) = YH(I) - YW(I)
      J=NLINV(I)+1
      IF (J .NE. 1) GO TO 80
*  IF YRDOT(1) NEEDED, EXTRAPOLATE YHDOT FROM STATIONS .5 AND 1.5
      YRDOT(I) = 1.5 * YHDOT(1) - .5 * YHDOT(2) - YWDOT(I)
      GO TO 110
 80   IF (J .NE.NHSP1) GO TO 90
      YRDOT(I) = -.5 * YHDOT(J-2) + 1.5 * YHDOT(J-1) - YWDOT(I)
      GO TO 110
   90 YRDOT(I) = .5 * (YHDOT(J-1) + YHDOT(J)) - YWDOT(I)
 110  CONTINUE
*
*  COMPUTE THE VERTICAL VELOCITY OF THE WAVE RELATIVE TO THE SHIP
      VR(1)=YRDOT(1)-ZBY4DXI(1)*(YR (2)-YR(1))
      VR(NCALC)=YRDOT(NCALC)-ZBY4DXI(NCALC)*(YR(NCALC)-YR(NCLCM1) )
      DO 135 I=2,NCLCM1
 135  VR(I)=YRDOT(I)-ZBY4DXI(I)*(YR(I+1)-YR(I-1))
*
```

Table 12 (Continued)

```
*   BEGIN ACCUMULATING THE HYDRODYNAMIC FORCE
*
*   LOOP TO 400 FOR BUOYANCY AND ADDED MASS AT EACH CALCULATED FORCE
*   HALF STATION
*
      DO 400 I=1,NCALC
      HFTEMP(I)=-BUOYSPG(I)*YR(I)
      JJGROUP=0
      IF (CONTROL(3) .NE. YES .OR. NL(I) .EQ. 0 .OR. NL(I) .EQ. 2)
     1 GO TO 300
*   COMPUTE NON-LINEAR BUOYANCY IF SELECTED.
      IB=IB+1
      JGROUP=JJGROUP=IBNL(1,IB)
      JUMP=1
  200 KCOND=1
      GO TO (280,220,240,220) ITYPE(JGROUP)
  220 YDELT=YR(I)-YCRIT(JGROUP)
      IF (YDELT .GT. 0) KCOND=2
      IF (ITYPE(JGROUP) .EQ. 2) GO TO 270
      IPLUS=2
      GO TO 250
  240 IPLUS=1
  250 IF (VR(I) .GT. VCRIT(JGROUP   )) KCOND=KCOND+IPLUS
  270 GO TO (280,380)JUMP
  280 IAD=IBNL(KCOND+1,IB)
      HFTEMP(I)=HFTEMP(I)+NOLEX(IBSTORE(IAD),BSTORE(IAD+1),YR(I))
  300 VMTEMP(I)=ADD MS(I)
      IF (CONTROL(3).NE.YES.OR.NL(I).EQ. 0.OR.NL(I).EQ.1) GO TO 400
*   COMPUTE NON-LINEAR ADDED MASS IF SELECTED.
      IM=IM+1
      JGROUP=IMNL(1,IM)
      JUMP=2
      IF (JGROUP .NE. JJGROUP) GO TO 200
  380 IAD=IMNL(KCOND+1,IM)
      VMTEMP(I)=VMTEMP(I)+NOLEX(IMSTORE(IAD,1),AMSTORE(IAD+1,1)
     1 YR(I))
*   ADD THE NON-LINEAR ADDED MASS TERM TO HYDRO FORCE.
      HFTEMP(I)=HFTEMP(I)-VR(I)*YRDOT(I)*NOLEX(IMSTORE(IAD,2),
     1 AMSTORE(IAD+1,2),YR(I))
  400 CONTINUE
      IF (IND .NE. 1) GO TO 420
      WRITE(6,410) T,NEVAL
  410 FORMAT(6H1TIME ,F11.6,I16,10H FCN EVALS  )
   99 FORMAT(A9,(9E13.6))
      WRITE (6,99) TWH,YW
      WRITE (6,99) TWV,YWDOT
      WRITE (6,99) TYWDD,YWDD
      WRITE (6,99) TYR,YR
      WRITE (6,99) TYRDOT,YRDOT
      WRITE (6,99) TVR,VR
      WRITE (6,99) TAM,(VMTEMP(I),I=1,NCALC)
      WRITE (6,99) THF3,(HFTEMP(I),I=1,NCALC)
*
*   COMPUTE THE SPACE DERIVATIVE TERMS AND ADD TO THE HYDROFORCE
*
  420 DO 430 I=1,NCALC
      I1=I-1
      IF (I.EQ. 1) I1=1
      I2=I+1
      IF (I .EQ. NCALC) I2=NCALC
      HFTEMP(I)=HFTEMP(I)+Z8Y4DXI(I)*(VMTEMP(I)*(YRDOT(I2)-YRDOT(I1))
     1+VMTEMP(I2)*VR(I2)-VMTEMP(I1)*VR(I1))
  430 CONTINUE
      IF (IND .EQ. 1) WRITE(6,99) THF4,(HFTEMP(I),I=1,NCALC)
*
*   COMPUTE THE SMITH CORRECTION AND ITS ASSOCIATED TERM AND ADD TO THE
*   HYDROFORCE
*
      TEMP = ABS(CEL + SPEED)
      IF (TEMP .GT. 1.E-10)  GO TO 460
      TEMP = 0.
      GO TO 461
```

Table 12 (Continued)

```
 460 TEMP = (CEL / TEMP)**2                                              HDFRC
 461 CONTINUE                                                            HDFRC
     DO 470 I=1,NCALC                                                    HDFRC
470  HFTEMP(I)=HFTEMP(I)+(VMTEMP(I)+TEMP*SCF(I))*YWDD(I)                 HDFRC
     IF (IND .EQ. 1) WRITE(6,99) THF5,(HFTEMP(I),I=1,NCALC)              HDFRC
*                                                                        HDFRC
*  COMPUTE THE HYDRODAMPIING AND ADD TO THE HYDROFORCE                   HDFRC
*                                                                        HDFRC
     IF (CONTROL(4) .NE. YES) GO TO 640                                  HDFRC
     DO 620 I=1,NCALC                                                    HDFRC
620  HFTEMP(I)=HFTEMP(I)-HYDDAMP(I)*VR(I)                                HDFRC
     IF (IND .EQ. 1) WRITE(6,99) THF7,(HFTEMP(I),I=1,NCALC)              HDFRC
640  CONTINUE                                                            HDFRC
*                                                                        HDFRC
*  DISTRIBUTE THE HYDROFORCE AND VIRTUAL MASS TO ALL HALF STATIONS       HDFRC
*                                                                        HDFRC
     DO 690 J=1,NHSTNS                                                   HDFRC
     K=NAPPLY(J)                                                         HDFRC
     IF (K .EQ. 0) GO TO 685                                             HDFRC
     HF(J)=HFTEMP(K)                                                     HDFRC
     VRTMS(J)=SHPMS(J)+VMTEMP(K)                                         HDFRC
     GO TO 690                                                           HDFRC
685  HF(J)=0.                                                            HDFRC
     VRTMS(J)=SHPMS(J)                                                   HDFRC
690  CONTINUE                                                            HDFRC
     CALL SLAM (YRDOT,YR,YW,SMFRC,T,SMPTS)                               HDFRC
     DO 693 IIS=1,12                                                     HDFRC
     ISTN = IIS + 9                                                      HDFRC
 693 HF(ISTN) = HF(ISTN) + SMPTS(IIS)                                    HDFRC
 696 IF (IND .NE. 1) RETURN                                              HDFRC
     WRITE (6,99) SLM,((SMFRC(I,J),J=1,2),I=1,6)                         HDFRC
     WRITE (6,99) SMOTH,(SMPTS(I),I=1,12)                                HDFRC
     WRITE(6,99) THF6,(HF(J),J=1,NHSTNS)                                 HDFRC
     DO 700 I=1,NCALC                                                    HDFRC
     K=NLINV(I)                                                          HDFRC
700  VMTEMP(I)=VRT MS(K)                                                 HDFRC
     WRITE(6,99) TVM,(VMTEMP(I),I=1,NCALC)                               HDFRC
     RETURN                                                              HDFRC
     ENTRY HFCIN                                                         HDFRC
     OLD T=-1.E20                                                        HDFRC
     NHSP1=NHSTNS+1                                                      HDFRC
     NCLCM1=NCALC-1                                                      HDFRC
     IND = 0                                                             HDFRC
C    UBY4DXI = SPEED / (4*DXI)                                           HDFRC
     DO 25 I=1,NCALC                                                     HDFRC
     I1=I-1                                                              HDFRC
     IF (I.EQ. 1) I1=1                                                   HDFRC
     I2=I+1                                                              HDFRC
     IF (I.EQ. NCALC) I2=NCALC                                           HDFRC
25   ZBY4DXI(I)=SPEED/((NLINV(I2)-NLINV(I1))*DXI)                        HDFRC
     DO 63 J=1,NHSTNS                                                    HDFRC
     I=NAPPLY(J)                                                         HDFRC
     IF (NAPPLY(J) .EQ. 0) GO TO 62                                      HDFRC
     VRT MS(J)=SHP MS(J)+ADD MS(I)                                       HDFRC
     GO TO 63                                                            HDFRC
62   VRT MS(J)=SHP MS(J)                                                 HDFRC
63   CONTINUE                                                            HDFRC
     RETURN                                                              HDFRC
     END                                                                 HDFRC
```

Table 12 (Continued)

```
      SUBROUTINE SLAM (YRDOT,YR,YW,SMFRC,T,SMPTS)                       SLAM
C     THIS IS MULTI-SLAM SMOOTHING ROUTINE                              SLAM
      DIMENSION YRDOT(9),YR(9),YW(9)                                    SLAM
      DIMENSION SMFRC(6,2),SMPTS(12)                                    SLAM
      DIMENSION COEF(8,3),WIDTH(10),DRHGT(10)                           SLAM
      COMMON/CELER/CELT                                                 SLAM
C     THIS COMMON IS SAME AS CD1, EXCEPT DX REPLACES DXI                SLAM
      COMMON /HBLOK/ KAG(21),RINT(21),EI(20),SHPMS(20),DAMPC(20),       SLAM
     1DX,NEVAL,SPEED,BUOY SPG(9),SCF(9),ADDMS(9),VRTMS(20),IBEG,IEND    SLAM
      COMMON /PLTSLM/ RAWSLM,SSMPTS                                     SLAM
      COMMON /IHVSS/ IHVS                                               SLAM
      DATA ISTSLM/0/                                                    SLAM
C                                                                       SLAM
C     VARIABLE  DEFINITION  (SYMBOL)                                    SLAM
C     --------  ----------  --------                                  SLAM
C     BTANG  BUTTOCK ANGLE  (ALPHA)                                     SLAM
C     DRANG  DEAD RISE ANGLE  (BETA)                                    SLAM
C     TRANG  TRIM ANGLE  (TAU)                                          SLAM
C     PHI  IMPACT ANGLE  (PHI)                                          SLAM
C     THETA  WAVE ANGLE  (THETA)                                        SLAM
C     SINPHI  SIN OF ANGLE PHI                                          SLAM
C     COSPHI  COSINE OF ANGLE PHI                                       SLAM
C     TANPHI  TANGENT OF ANGLE PHI                                      SLAM
C     SINTHE  SIN OF ANGLE THETA                                        SLAM
C     COSTHE  COSINE OF ANGLE THETA                                     SLAM
C     TANTHE  TANGENT OF ANGLE THETA                                    SLAM
C     SINAPH  SIN OF ANGLE (ALPHA + PHI)                                SLAM
C     COSAPH  COSINE OF ANGLE (ALPHA + PHI)                             SLAM
C     TANAPH  TANGENT OF ANGLE (ALPHA + PHI)                            SLAM
C     SINTAL  SIN OF ANGLE (TAU + ALPHA)                                SLAM
C     COSTAL  COSINE OF ANGLE (TAU + ALPHA)                             SLAM
C     TANTAL  TANGENT OF ANGLE (TAU + ALPHA)                            SLAM
C     COSBET  COSINE OF ANGLE BETA                                      SLAM
C     TANBET  TANGENT OF ANGLE BETA                                     SLAM
C     TANXI  TANGENT OF ANGLE XI                                        SLAM
C     SECBEH  SECANT OF ANGLE BETA(SUB EH)                              SLAM
C     WIDTH(10)  WIDTH OF SLAM REGION  (SMALL B)                        SLAM
C     DRHGT(10)  HEIGHT OF DEAD RISE  (B)                               SLAM
C     COEF(7,3)  POLYNOMIAL ORDERS + COEFFICIENTS                       SLAM
C     DRAFT  SHIPS DRAFT  (D)                                           SLAM
C     FACTK  SLAM PRESSURE FACTOR  (K(SUB XI))                          SLAM
C     SMLGT  LENGTH OF SLAM REGION  (DELTA SMALL L)                     SLAM
C     VLNIMP  IMPACT VELOCITY - NORMAL  (V(SUB N))                      SLAM
C     VLTIMP  IMPACT VELOCITY - TANGENTIAL  (V(SUB T))                  SLAM
C     VR(9)  RELATIVE VERTICAL VELOCITY  (V(SUB R))                     SLAM
C     YR(9)  RELATIVE VERTICAL DISPLACEMENT  (Y(SUB R))                 SLAM
C     YW(9)  WAVE HEIGHT  (Y(SUB W))                                    SLAM
C     SMSTRT  POINT WHERE SLAM STARTS  (X(SUB 0))                       SLAM
C     DX  STATION LENGTH  (DELTA X)                                     SLAM
C     DX2  2 TIMES STATION LENGTH  (2*DX)                               SLAM
C     DX10  10 TIMES STATION LENGTH  (10*DX)                            SLAM
C     SMEND  POINT WHERE SLAM ENDS  (X(SUB E))                          SLAM
C     XS  DISTANCE FROM THE STERN AT WHICH THE SLAM FORCE SHOULD BE APPL SLAM
C     SMWGT1  WEIGHTS USED TO DISTRIBUTE THE SLAM FORCE  (W(SUB S+))    SLAM
C     SMWGT2  WEIGHTS USED TO DISTRIBUTE THE SLAM FORCE  (W(SUB S-))    SLAM
C     SMFRC(6,2)  SLAM FORCES  (F(SUB S))                               SLAM
C     DENS  MASS DENSITY OF WATER  (RHO)                                SLAM
C     DNSBY2  MASS DENSITY OF WATER DIVIDED BY 2                        SLAM
C     EFIMPA  EFFECTIVE IMPACT ANGLE  (XI)                              SLAM
C     KSLAM  NUMBER OF THE FIRST OF THE TWO STATIONS AT WHICH SLAM FORCE SLAM
C            IS APPLIED  (K(SUB S))                                     SLAM
C     NEND  NUMBER OF STATION CLOSEST TO X(SUB E)  (N(SUB E))           SLAM
C                                                                       SLAM
C                                                                       SLAM
C********************************************************************** SLAM
C*    PROBABLY SHOULD BE MODIFIED SO THAT SLAM OCCURS IF              * SLAM
C*    SHOULDER IS ABOVE WATER, RATHER THAN KEEL.                      * SLAM
C********************************************************************** SLAM
C                                                                       SLAM
C                                                                       SLAM
```

Table 12 (Continued)

```
C
C     TEST IF THIS IS FIRST CALL TO SLAM
C         IF SO, SET  SMFRC = 0.
C
      IF (ISTSLM .NE. 0)  GO TO 10
      DO 5 I=1,2
      DO 5 J=1,6
    5 SMFRC(J,I) = 0.
      RAWSLM = 0.
      ISTSLM = 1
C
C
C     DETERMINE WHETHER TO START WITH TYPE1 OR TYPE2 SLAM
C
   10 CONTINUE
C
C     CALL SLAM ON EVERY 5TH. (OR MULTIPLE OF) EVALUATION
      RAWSLM = 0.
      IIHVS = IHVS - 4
      KNEVAL = 5
      IF (IHVS .GE. 5)  KNEVAL = KNEVAL * 2**IIHVS
      IF (MOD(NEVAL,KNEVAL) .NE. 0)  GO TO 301
C
      IF (YR(9) .LT. DRAFT)  GO TO 800
  804 NSLM = 1
      ICASE = 1
      SGNFTR = 1.
      ISGNF = IFIX(SGNFTR)
  801 DO 290  KBWD=1,4
      KFWD = 9 - KBWD
      TEMP = YR(KFWD) - DRAFT
      IF (TEMP*SGNFTR .GE. 0.)  GO TO 290
C
C     FIND THE POINT WHERE THE KEEL TOUCHES THE WAVE SURFACE
C
      TANPHI = (YR(KFWD+1) - YR(KFWD)) / DX2
      TEMP = -(TEMP/TANPHI)
      SMSTRT = (2*KFWD+1) * DX + TEMP
      TEMP = TEMP / DX2
C
C     COMPUTE THE VERTICAL VELOCITY A, THE POINT WHERE THE KEEL TOUCHES
C     THE WAVE SURFACE.  IF THIS POINT NOT MOVING DOWN, LOOK FOR NEXT
C     SLAM TYPE.
C
      VRTVLC = TEMP * YRDOT(KFWD+1) + (1.-TEMP) * YRDOT(KFWD)
      IF (ICASE .EQ. 2) GO TO 803
      IF (VRTVLC .GE. 0.)  GO TO 300
      GO TO 900
  803 IF (VPTVLC .GE. 0.) GO TO 804
      GO TO 900
C
C     DETERMINE THE LENGTH OF SLAM REGION
C
  900 SMLGT = 0.
      SMEND = SMSTRT
      NEND = 20 * ISGNF
      DO 25 J=1,10
      M = IFIX(SMEND/DX + .5) - 10
      IF ( (M-NEND) * ISGNF .GE. 0) GO TO 30
      NEND = M
      IF (ISGNF .EQ. 1)  SMLGT = DRHGT(NEND) / TANPHI
      IF (ISGNF .EQ. -1)  SMLGT = SMLGT - ((DRHGT(NEND) + SMLGT *
     1                                  TANPHI) / (BSLP - TANPHI))
      SMEND = SMSTRT - SMLGT
      IF (SMEND .GE. DX20) GO TO 29
      IF (SMEND .LT. DX105) GO TO 28
   25 CONTINUE
      GO TO 30
C
C     THE SLAM FORCE SHOULD BE APPLIED
C     AT THE DISTANCE X(S) FROM THE STERN
C
```

Table 12 (Continued)

```
   28 SMEND = DX105
      GO TO 295
   29 SMEND = DX20
  295 SMLGT = SMSTRT - SMEND
   30 XS = (5.*SMSTRT + 4.*SMEND) / 9.
C
C     THE SLAM FORCE WILL BE APPLIED AT STATIONS NUMBERED K(S) AND K(S)+
C     USING WEIGHTS W(S+) AND W(S-)
C
      SMWGT1 = XS/DX + .5
      KSLAM = IFIX(SMWGT1)
      SMWGT1 = SMWGT1 - FLOAT(KSLAM)
      SMWGT2 = 1. - SMWGT1
C
C     DETERMINE THE EFFECTIVE IMPACT ANGLE XI
C
      COSPHI = 1. / SQRT(1. + TANPHI**2)
      SINPHI = TANPHI * COSPHI
      SINAPH = SINAL * COSPHI + COSAL * SINPHI
      COSAPH = COSAL * COSPHI - SINAL * SINPHI
      TANAPH = SINAPH / COSAPH
      TEMP1 = TANBET / (SINPHI + TANAL * COSPHI)
      SECBEH = SQRT(1. + TEMP1**2) * SGNFTR
      COSBEH = 1. / SECBEH
      TANXI = TANAPH * SECBEH
      EFIMPA = ATAN(TANXI)
C
C     CALCULATE THE TANGENTIAL AND NORMAL IMPACT VELOCITIES
C
      TANTHE = (YW(KFWD+1) - YW(KFWD)) / DX2
      COSTHE = 1. / SQRT(1. + TANTHE**2)
      SINTHE = TANTHE * COSTHE
      SINTAL = SINAPH * COSTHE  +  COSAPH * SINTHE
      COSTAL = COSAPH * COSTHE - SINAPH * SINTHE
      TEMP = VRTVLC * COSTAL  +  (SPEED+CELT) * SINTAL
      VLNIMP = TEMP * COSAPH / COSTAL**2
      VLTIMP = TEMP  * SINAPH
C
C     CALCULATE THE FACTOR K(XI)  (FACTK)
C
      IF (EFIMPA .GE. .0383972) GO TO 85
      FACTK = POLY(COEF(1,1),EFIMPA)
      GO TO 110
   85 IF (EFIMPA .GE. .191986) GO TO 90
      FACTK = POLY(COEF(1,2),EFIMPA)
      GO TO 110
   90 IF (EFIMPA .GE. .349066) GO TO 100
      FACTK = POLY(COEF(1,3),EFIMPA)
      GO TO 110
C     .76856471/288. = .00266862747
C     .00266862747 * 2.4674 = .00658457141
  100 FACTK = .00266862747 + .00658457141 / TANXI **2
C
C     CALCULATE THE SLAM PRESSURE
C
  110 PRESS = FACTK * VLNIMP**2 * 144.
      PRESS = PRESS + (.5 * VLTIMP**2 * COSBEH)
      PRESS = PRESS * DENS / 2240.
C
C     CALCULATE AND DISTRIBUTE THE SLAM FORCE
C
      TEMP = COSPHI * COSTHE - SINPHI * SINTHE
      TEMP = .5 * PRESS * DRHGT(NEND) * SGNFTR * SMLGT * TEMP
C     CHANGE FORCE TO FORCE/FOOT
      TEMP = TEMP/DX
C
      TEMP=5.*TEMP
C     TEMPORARY INCREASE IN SLAM FOR DISPLAY PURPOSES
```

Table 12 (Continued)

```
C
      RAWSLM = RAWSLM + TEMP
      TIMCUT = YR(KFWD) - ORHGT(KSLAM-10)
      SMFRC(1,NSLM) = FLOAT(KSLAM)
      SMFRC(2,NSLM) = SMWGT2 * TEMP
      SMFRC(3,NSLM) = SMWGT1 * TEMP
      SMFRC(4,NSLM) = YR(KFWD)
      SMFRC(5,NSLM) = TIMCUT
      SMFRC(6,NSLM) = KFWD
      IF (ICASE .EQ. 1)  GO TO 300
      NSLM = NSLM - 1
      ICASE = 1
      SGNFTR = 1.
      ISGNF = IFIX(SGNFTR)
  290 CONTINUE
      IF (ICASE .EQ. 1)  GO TO 300
      DO 299  NNSLM=1,2
      IF (SMFRC(1,NNSLM) .EQ. 0.)  GO TO 299
      OLDKFW = SMFRC(6,NNSLM)
      IF (YR(OLDKFW) .LE. SMFRC(5,NNSLM) )  GO TO 802
      TIMFACT = (SMFRC(5,NNSLM) - YR(OLDKFW)) /
     1           (SMFRC(5,NNSLM) - SMFRC(4,NNSLM))
      SMFRC(2,NNSLM) = SMFRC(2,NNSLM) * TIMFACT
      SMFRC(3,NNSLM) = SMFRC(3,NNSLM) * TIMFACT
      SMFRC(4,NNSLM) = YR(OLDKFW)
  299 CONTINUE
      GO TO 300
  802 SMFRC(1,NNSLM) = 0.
      IF (NNSLM .EQ. 1)  GO TO 299
      GO TO 300
  800 NSLM = 2
      ICASE = 2
      SGNFTR = -1.
      ISGNF = IFIX(SGNFTR)
      GO TO 801
  300 CALL SSMOOTH (SMFRC,SMPTS,I1281)
  301 RETURN
C
C
      ENTRY INSMIN
C     INSMIN SETS DEFAULTS
      RETURN
C
C
      ENTRY INSMDT
C     INSMDT "READS" CARDS
      READ (5,1000) DRAFT,DENS,BTANG,DRANG
 1000 FORMAT (4F10.0)
      READ (5,1010)((COEF(I,1),I=1,8),(COEF(I,2),I=1,8),
     1(COEF(I,3),I=1,8),(WIDTH(I),I=1,10))
 1010 FORMAT (8F10.0)
      READ (5,1020) BSLP,I1281
 1020 FORMAT (F10.0,I10)
      RETURN
C
C
      ENTRY INSMFIN
C     INSMFIN PRE-COMPUTES AND WRITES DATA
C
      WRITE (6,1990)
 1990 FORMAT (1H1,10HINPUT DATA)
      WRITE (6,2000) DRAFT,DENS,BTANG,DRANG,DX
 2000 FORMAT (1H0,7X,22HDRAFT = SHIPS DRAFT = ,E13.6,1X,2HFT/
     11H ,7X,31HDENS = MASS DENSITY OF WATER = ,E13.6,1X,12HSLUGS/CU.FT.
     2/1H ,7X,31HBTANG = ALPHA, BUTTOCK ANGLE = ,E13.6,1X,7HDEGREES/
     31H ,7X,31HDRANG = BETA, DEADRISE ANGLE = ,E13.6,1X,7HDEGREES/
     41H ,7X,22HDX = STATION LENGTH = ,E13.6,1X,2HFT)
      WRITE(6,2005) BSLP
 2005 FORMAT (1H ,7X,38HBSLP = SLOPE RELATED TO WIDTH ARRAY = ,E13.6)
```

Table 12 (Continued)

```
C     CHANGE ALPHA AND BETA TO RADIANS
      BTANG = BTANG * .01745329252
      DRANG = DRANG * .01745329252
C
      SINAL = SIN(BTANG)
      COSAL = COS(BTANG)
      TANAL = TAN(BTANG)
      COSBET = COS(DRANG)
      TANBET = TAN(DRANG)
      DX2 = 2. * DX
      DX10 = 10. * DX
      DX105 = 10.5 * DX
      DX20 = 20. * DX
      DNSBY2 = DENS/2.
      DO 1 K=1,10
    1 DRHGT(K) = WIDTH(K) * TANBET
      WRITE (6,2010)
 2010 FORMAT (1H-,17HPRE-COMPUTED DATA)
      WRITE (6,2020) SINAL,COSAL,TANAL,COSBET,TANBET
 2020 FORMAT (1H0,7X,24HSINAL = SINE OF ALPHA = ,E13.6/
     11H ,7X,26HCOSAL = COSINE OF ALPHA = ,E13.6/
     21H ,7X,27HTANAL = TANGENT OF ALPHA = ,E13.6/
     31H ,7X,26HCOSBET = COSINE OF BETA = ,E13.6/
     41H ,7X,27HTANBET = TANGENT OF BETA = ,E13.6)
      WRITE (6,2030) DX2,DX10,DNSBY2
 2030 FORMAT (1H ,7X,28HDX2 = 2. * STATION LENGTH = ,E13.6,1X,2HFT/
     11H ,7X,30HDX10 = 10. * STATION LENGTH = ,E13.6,1X,2HFT/
     21H ,7X,38HDNSBY2 = MASS DENSITY OF WATER / 2. = ,E13.6,1X,
     312HSLUGS/CU.FT.)
      WRITE (6,2040)
 2040 FORMAT (1H0,14X,18HHEIGHT OF DEADRISE,14X,20HWIDTH OF SLAM REGION/
     11H ,18X,9HDRHGT(10),24X,9HWIDTH(10))
      WRITE (6,2050) ((DRHGT(I),WIDTH(I)),I=1,10)
 2050 FORMAT (10(1H ,14X,E13.6,1X,2HFT,17X,E13.6,1X,2HFT/))
      WRITE (6,2060)
 2060 FORMAT (1H-,14X,9HCOEF(8,1),14X,9HCOEF(8,2),14X,9HCOEF(8,3))
      WRITE (6,2070) ((COEF(I,1),COEF(I,2),COEF(I,3)),I=1,8)
 2070 FORMAT (8(1H ,11X,E13.6,10X,E13.6,10X,E13.6/))
      RETURN
      END
```

```
      SUBROUTINE SSMOOTH (SMFRC,SMPTS,I1281)
      DIMENSION SMFRC(6,2), SSMARY(12,1281), ACOEF(1281), SMPTS(12)
      DIMENSION KSTATN(12)
      COMMON /IHVSS/ IHVS
      COMMON /PLTSLM/ RAWSLM,SSMPTS
      DATA PI /3.14159265/
      DATA ISTSSM /0/
C
C     IF THIS IS FIRST CALL TO SSMOOTH, DO SOME INITIALIZING
      IF (ISTSSM .NE. 0)  GO TO 1010
      SUMCF1 = 0.
      SUMCF2 = 0.
      SUMCF4 = 0.
      SUMCF8 = 0.
      DO 1001 I=1,I1281
      ACOEF(I) = SIN(PI * ((FLOAT(I)-1.) / (FLOAT(I1281)-1) ))
 1001 SUMCF1 = SUMCF1 + ACOEF(I)
      SUMCF1 = 1. / SUMCF1
      DO 1021 I=1,I1281,2
 1021 SUMCF2 = SUMCF2 + ACOEF(I)
      SUMCF2 = 1. / SUMCF2
      DO 1041 I=1,I1281,4
 1041 SUMCF4 = SUMCF4 + ACOEF(I)
      SUMCF4 = 1. / SUMCF4
      DO 1081 I=1,I1281,8
 1081 SUMCF8 = SUMCF8 + ACOEF(I)
      SUMCF8 = 1. / SUMCF8
      DO 1003 I=1,12
 1003 SMPTS(I) = 0.
      DO 1004 I=1,12
 1004 KSTATN(I) = 0
      SSMPTS = 0.
      KB = 0
      KE = 0
      KBB = 1
      IHVSLST = 1
      ISTSSM = 1
C
 1010 CONTINUE
C
C     DO NOT SMOOTH IF THERE HAS NOT BEEN ANY SLAMMING YET
      IF (SMFRC(2,1) .EQ. 0.  .AND.  SMFRC(2,2) .EQ. 0.)  GO TO 5000
C
C
      IF (IHVS .GE. 1)  GO TO 1020
      WRITE(6,7) IHVS
    7 FORMAT (1H1,E13.6,*....................IHVS IS BUM.............*)
      GO TO 5000
 1020 IF (SMFRC(1,1) .NE. 0.  .AND.
     1   (SMFRC(1,1) .LT. 10.  .OR.  SMFRC(1,1) .GT. 20.))  GO TO 1025
      IF (SMFRC(1,2) .EQ. 0.  .OR.
     1   (SMFRC(1,2) .GE. 10.  .AND.  SMFRC(1,2) .LE. 20.))  GO TO 1030
 1025 WRITE (6,8) SMFRC(1,1),SMFRC(1,2)
    8 FORMAT (1H1,*SMFRC(1,1)=*,E13.6,5X,*SMFRC(1,2)=*,E13.6,*..........
     1.............................................BAD*)
      GO TO 5000
C
 1030 CONTINUE
      IF (IHVS .GE. 4)  GO TO 4
      IF (IHVS-2)  1,2,3
    1 IMANY = 8
C     WHOLE TIME INCREMENT, IHVS=1, USE EVERY 8TH. LOCATION
      TK = SUMCF8
      GO TO 1040
C
    2 IMANY = 4
C     1/2 TIME INCREMENT, IHVS=2, USE EVERY 4TH. LOCATION
      TK = SUMCF4
      GO TO 1040
C
    3 IMANY = 2
```

```
C     1/4 TIME INCREMENT, IHVS=3, USE EVERY OTHER LOCATION                SSMOOTH
      TK = SUMCF2                                                         SSMOOTH
      GO TO 1040                                                          SSMOOTH
C                                                                         SSMOOTH
    4 IMANY = 1                                                           SSMOOTH
C     1/8 OR SMALLER TIME INCREMENT, IHVS=4 OR MORE, USE EVERY LOCATION   SSMOOTH
      TK = SUMCF1                                                         SSMOOTH
C                                                                         SSMOOTH
 1040 CONTINUE                                                            SSMOOTH
C                                                                         SSMOOTH
C                                                                         SSMOOTH
C     PUT THE SLAM FORCE INTO THE ARRAY TO BE SMOOTHED.                   SSMOOTH
C     REPEAT VALUE AS MANY TIMES NECESSARY FOR THE TIME INCREMENT.        SSMOOTH
C                                                                         SSMOOTH
C     A SLAM CAN OCCUR ON STATION NUMBERS 10 THROUGH 20                   SSMOOTH
C     IF SLAMMING STATION WAS  10  THEN ISTATN AND K ARE  1               SSMOOTH
C                              11 ......................  2               SSMOOTH
C                              12 ......................  3               SSMOOTH
C                              13 ......................  4               SSMOOTH
C                              14 ......................  5               SSMOOTH
C                              15 ......................  6               SSMOOTH
C                              16 ......................  7               SSMOOTH
C                              17 ......................  8               SSMOOTH
C                              18 ......................  9               SSMOOTH
C                              19 ...................... 10               SSMOOTH
C                              20 ...................... 11               SSMOOTH
C                                 ...................... 12               SSMOOTH
C                                                                         SSMOOTH
      DO 100 ITYPE=1,2                                                    SSMOOTH
      ISTATN = IFIX (SMFRC(1,ITYPE) - 9.)                                 SSMOOTH
      IF (SMFRC(1,1) .EQ. 0.)  ISTATN = -1                                SSMOOTH
      IF (SMFRC(1,2) .EQ. 0.  .AND.  ITYPE .EQ. 2)  GO TO 150             SSMOOTH
      IF (ITYPE .EQ. 2)  GO TO 96                                         SSMOOTH
      KB = KE + 1                                                         SSMOOTH
C     CHECK IHVS AS IT COMES IN.  IF IT IS GREATER THAN LAST TIME,        SSMOOTH
C     THEN THE INTERVAL HAS BEEN HALVED AND THE SSMARY MUST BE            SSMOOTH
C     REWRITTEN STARTING AT THE PREVIOUS PLACE (THAT IS KBB).             SSMOOTH
      IF (IHVS .GT. IHVSLST)  KB = KBB                                    SSMOOTH
      KE = KB + IMANY - 1                                                 SSMOOTH
      DO 99 K=1,12                                                        SSMOOTH
      DO 99 III=KB,KE                                                     SSMOOTH
      IIIMOD = MOD(III,I1281)                                             SSMOOTH
      IF (IIIMOD .EQ. 0)  IIIMOD = I1281                                  SSMOOTH
      IF (K .EQ. ISTATN  .OR.  K .EQ. ISTATN+1)  GO TO 90                 SSMOOTH
      SSMARY(K,IIIMOD) = 0.                                               SSMOOTH
      GO TO 99                                                            SSMOOTH
   90 IF (K .EQ. ISTATN+1)  GO TO 95                                      SSMOOTH
      SSMARY(K,IIIMOD) = SMFRC(2,ITYPE)                                   SSMOOTH
      GO TO 99                                                            SSMOOTH
   95 SSMARY(K,IIIMOD) = SMFRC(3,ITYPE)                                   SSMOOTH
   99 CONTINUE                                                            SSMOOTH
      IF (ISTATN .LT. 1)  GO TO 100                                       SSMOOTH
      KSTATN (ISTATN  ) = I1281                                           SSMOOTH
      KSTATN (ISTATN+1) = I1281                                           SSMOOTH
  100 CONTINUE                                                            SSMOOTH
C                                                                         SSMOOTH
   96 ISTATN = IFIX (SMFRC(1,2) - 9.)                                     SSMOOTH
      DO 98 III=KB,KE                                                     SSMOOTH
      IIIMOD = MOD(III,I1281)                                             SSMOOTH
      IF (IIIMOD .EQ. 0)  IIIMOD = I1281                                  SSMOOTH
      SSMARY(ISTATN  ,IIIMOD) = SSMARY(ISTATN  ,IIIMOD) + SMFRC(2,2)      SSMOOTH
   98 SSMARY(ISTATN+1,IIIMOD) = SSMARY(ISTATN+1,IIIMOD) + SMFRC(3,2)      SSMOOTH
      KSTATN (ISTATN  ) = I1281                                           SSMOOTH
      KSTATN (ISTATN+1) = I1281                                           SSMOOTH
  150 CONTINUE                                                            SSMOOTH
      KBB = KB                                                            SSMOOTH
C                                                                         SSMOOTH
C                                                                         SSMOOTH
C     INITIALIZE THE SMOOTHED POINTS.                                     SSMOOTH
      DO 310 I=1,12                                                       SSMOOTH
  310 SMPTS(I) = 0.                                                       SSMOOTH
```

Table 12 (Continued)

```
C
C     DETERMINE THE SMOOTHED POINTS BY 'FLIP-FLOP' MULTIPLICATION
      KBIC = MAX0(1,KE-(I1281-1))
      DO 301 K=1,12
      IF (KSTATN(K) .LE. 0)  GO TO 301
      DO 300 IC=KBIC,KE,IMANY
      ICMOD = MOD(IC,I1281)
      IF (ICMOD .EQ. 0)  ICMOD = I1281
      IB = KE - IC + 1
      SMPTS(K) = SMPTS(K) + SSMARY(K,IB) * ACOEF(ICMOD)
  300 CONTINUE
      SMPTS(K) = SMPTS(K) * TK
      KSTATN(K) = KSTATN(K) - IMANY
  301 CONTINUE
C
C     SUM ALL THE SMOOTHED POINTS FOR PLOTTING ?????????????????????????
      SSMPTS = 0.
      DO 350 K=1,11,2
  350 SSMPTS = SSMPTS + SMPTS(K) + SMPTS(K+1)
C
      IMVSLST = IHVS
 5000 RETURN
      END
```

```
      SUBROUTINE SEA GEN(YW,YWDOT,YWDD,CEL,T)
      DIMENSION YW(9),YWDOT(9),YWDD(9),ACOSX(3,9),ASINX(3,9),
     1          COSOMT(3),SINOMT(3),AMP(3),CK(3),TAU(3),XK(9),
     2          OMEGA(3),OMG SQ(3)
      COMMON/FBLOK/FIELDA(6),IF(6)
      COMMON/INPT2/J,ISFIELD,BUFFER(60),ICFIELD
      COMMON /HBLOK/ KAG(21),RINT(21),EI(20),SHPMS(20),DAMPC(20),
     1 DXI,NEVAL,SPEED,BUOY SPG(9),SCF(9),ADDMS(9),VRTMS(20),IBEG,IEND
      DATA T1,              T2,              T3              /
     1     -28.40945342,    -25.51319346,    17.02731343     /,
     2     DIFF,            AMP                              /
     3     .2627819036,     3.270009343,     2*5.5261579     /
      DO 130 I = 1,3
      COSOMT(I) = COS(OMEGA(I)*T)
  130 SINOMT(I) = SIN(OMEGA(I)*T)
      DO 160 K = 1,9
      TXK = T + XK(K)
      IF (TXK.GT.T1) GO TO 120
      YW(K) = 0.
      YWDOT(K) = 0.
      YWDD(K) = 0.
      GO TO 160
  120 IF (TXK.GT.T3) GO TO 140
      YW(K) = ASINX(1,K) * COSOMT(1) + ACOSX(1,K) * SINOMT(1) + DIFF
      YWDOT(K)=OMEGA(1)*(ACOSX(1,K)*COSOMT(1)-ASINX(1,K)*SINOMT(1))
      YWDD(K)= -OMG SQ(1)*YW(K)
      IF (TXK.GT.T2) GO TO 160
      YW(K) = C1 * (C2+YW(K))
      YWDOT(K) = C1 * YWDOT(K)
      YWDD(K) = C1 * YWDD(K)
      GO TO 160
  140 YW(K) = 0.
      YWDOT(K) = 0.
      YWDD(K) = 0.
      DO 150 I = 2,3
      TEMP=ACOSX(I,K)*COSOMT(I)-ASINX(I,K)*SINOMT(I)
      YW(K) = YW(K) + TEMP
      YWDOT(K) = YWDOT(K)-OMEGA(I)*(ASINX(I,K)*COSOMT(I)
     1                            +ACOSX(I,K)*SINOMT(I))
  150 YWDD(K) = YWDD(K) - OMG SQ(I)*TEMP
  160 CONTINUE
      CEL = CELT
      RETURN
      ENTRY INSGFIN
      DO 210 I = 1,3
      OMG SQ(I) = OMEGA(I)**2.
  210 CK(I) = .011*OMEGA(I)
      CELT = (1 - .011*SPEED) / .011
      TEMP = 2.5
      DO 250 K = 1,9
      IF(K .EQ. 1)        GO TO 230
      TEMP = 1. + 2.*FLOAT(K)
  230 XK(K) = DXI*TEMP
      DO 240 I = 1,3
      TEMP = CK(I)*XK(K) + TAU(I)
      ACOSX(I,K) = AMP(I)*COS(TEMP)
  240 ASINX(I,K) = AMP(I)*SIN(TEMP)
*
*           FOR LATER, XK(K) (=K*XI) IS USED TO SELECT W1 OR W2.
      XK(K) = .011*XK(K)
  250 CONTINUE
      C1 = (AMP(1)+DIFF)/(2.*AMP(1))
      C2 = AMP(1) - DIFF
      WRITE (6,10)
   10 FORMAT (1H-,*SEAGEN IS USING DISCRETE WAVE TRAIN*)
      RETURN
      ENTRY INSEGIN
      RETURN
      ENTRY INSEGOT
      CALL FIELOS(52528,6,BUFFER)
      DO 310 I = 1,3
      TAU(I) = FIELDA(I + 3)
  310 OMEGA(I) = FIELDA(I)
      RETURN
      END
```

```
      SUBROUTINE SEAGEN (YW,YWDOT,YWDD,CEL,T)
C
C     CEL = CELERITY, WAVE SPEED
C     DELTA = (GRAVITY * PI)/(CEL + SPEED) = TIME(SUB 1) - TIME(SUB 0)
C     X(SUB 1,...,9) = ARRAY OF THE 9 STATIONS OF THE SHIP
C     XK(I) = K * X(I)
C     XK(1,...,9) IS TIME(SUB 0) FOR 9 STATIONS OF THE SHIP
C     OMEGAW = FREQUENCY OF THE WAVE
C     OMGWSQ = OMEGAW**2
C     OMEGAE = FREQUENCY OF ENCOUNTER
C     OMGESQ = OMEGAE**2
C     SINOMT = SIN(OMEGAE*TIME) = SIN OF THE ANGLE WITH TIME
C     COSOMT = COS(OMEGAE*TIME) = COSINE OF THE ANGLE WITH TIME
C     ASINX = WAVE HEIGHT * SIN(K * X(SUB I))
C     ACOSX = WAVE HEIGHT * COS(K * X(SUB I))
C     SPEED = U
C     TIME = T
C     GRAV = GRAVITY = 32.23
C     YW = 1/OMEGAW
C     YWDOT = FIRST DERIVATIVE OF 1/OMEGA = VELOCITY OF WAVE HEIGHT
C                                    AT SPECIFIC TIME + LOCATION
C     YWDD = SECOND DERIVATIVE OF 1/OMEGA = ACCELERATION OF WAVE HEIGHT
C                                    AT SPECIFIC TIME + LOCATION
C     WVLGT = LAMBDA = WAVE LENGTH
C     PI = 3.14159265
C     WVHGT = W = WAVE AMPLITUDE, WAVE HEIGHT
C
      COMMON/CELER/CELT
      COMMON/FBLOK/FA(6),IF(6)
      COMMON/INPT2/J,ISFIELD,BUFFER(60),ICFIELD
      COMMON /HBLOK/ KAG(21),RINT(21),EI(20),SHPMS(20),DAMPC(20),
     1 DXI,NEVAL,SPEED,BUOY SPG(9),SCF(9),ADDMS(9),VRTMS(20),IBEG,IEND
      DIMENSION X(9),XK(9),ASINX(9),ACOSX(9),YW(9),YWDOT(9),YWDD(9)
      REAL K
      DATA GRAV/32.23/,PI/3.14159265/
      DATA X/2.5,5.,7.,9.,11.,13.,15.,17.,19./
C
      CEL = CELT
      OMGTIM = OMEGAE * T
      SINOMT = SIN(OMGTIM)
      COSOMT = COS(OMGTIM)
      DO 400 I=1,9
      IF (WVHGT.EQ.0.) GO TO 210
      IF (OMGTIM.GT.XK(I)) GO TO 230
  210 YW(I) = 0.
      YWDOT(I) = 0.
      YWDD(I) = 0.
      GO TO 400
  230 YW(I) = ASINX(I) * COSOMT + ACOSX(I) * SINOMT
      YWDOT(I) = OMEGAE * (ACOSX(I) * COSOMT - ASINX(I) * SINOMT)
      YWDD(I) = -(OMGESQ * YW(I))
      IF (OMGTIM .GT. XK(I) + PI) GO TO 400
      YW(I) = .5 * (YW(I) - WVHGT)
      YWDOT(I) = .5 * YWDOT(I)
      YWDD(I) = .5 * YWDD(I)
  400 CONTINUE
      RETURN
C
      ENTRY INSEGIN
C     INSEGIN SETS DEFAULTS
      OMEGAW = 0.
      WVHGT = 0.
      RETURN
C
      ENTRY INSEGOT
C     INSEGOT "READS" CARDS THAT ARE INPUT
      CALL FIELDS (50008,6,BUFFER)
      OMEGAW = FA(1)
      WVHGT = FA(2)
      RETURN
C
```

Table 12 (Continued)

```
      ENTRY INSGFIN
C     INSGFIN SETS UP DATA + WRITES OUT INPUT DATA
      OMGWSQ = OMEGAW ** 2
      OMEGAE = OMEGAW + (SPEED * OMGWSQ / GRAV)
      OMGESQ = OMEGAE**2
      WVLGT = 2. * PI * GRAV / OMGWSQ
      K = OMGWSQ / GRAV
      CELT = GRAV / OMEGAW
      DELTA = GRAV * PI / (CELT + SPEED)
      PIBY2 = PI/2.
      DO 20 I=1,9
      XK(I) = K * DXI * FLOAT(I) * 2.
      ASINX(I) = WVHGT * SIN(XK(I))
      ACOSX(I) = WVHGT * COS(XK(I))
   20 XK(I) = -(XK(I) - PIBY2)
  250 WRITE (6,1000) OMEGAW,WVHGT,CELT,OMEGAE,WVLGT
 1000 FORMAT (1H1,17HWAVE FREQUENCY = ,F7.3/
     1             15H WAVE HEIGHT = ,F7.3/
     2             12H CELERITY = ,F7.3/
     3             26H FREQUENCY OF ENCOUNTER = ,F7.3/
     4             15H WAVE LENGTH = ,F7.3)
      RETURN
C
      END
```

Table 12 (Continued)

```
      SUBROUTINE SEA GEN (HF,YWDOT,YWDD,CEL,T)
C$    DEBUG
C$    CALLS
C$    STORES(FREQ,WVHGH)
C
C**SEA GENERATOR
C
      DIMENSION HF(1),YWDOT(1),YWDD(1),CS1(10)
      COMMON/FBLOK/FIELDA(6) ,IF(6)
      COMMON /INPT2/J,ISFIELD,BUFFER(60),ICFIELD
      DATA CS1/.000,.174,.342,.5,.643,.766,.866,.94,.985,1./
      CEL=CELT
      DO 5 I=1,20
 5    HF(I)=0.
      TEMP=WVHGH*SIN(OMEGA*T)
      DO 10 I=1,5
 10   HF(I+5)=HF(16-I)=CS1(2*I)*TEMP
      RETURN
C
C**SET DEFAULTS
C
      ENTRY INSEGIN
      FREQ=WVHGH=0.
      RETURN
C
C** ACCEPT DATA
C
      ENTRY INSEGOT
      CALL FIELDS(50008,6,BUFFER)
      FREQ=FIELDA(1)
      WVHGH=FIELDA(2)
      RETURN
C
C**DATA ECHO
C
      ENTRY INSGFIN
      OMEGA=2.*3.14159265358 98*FREQ
      CELT=32.23/OMEGA
      OMEGA2=OMEGA**2
      WRITE(6,90) FREQ,WV HGH
 90   FORMAT(1H-,*SEA DATA (CASE 1)*//T5,*FREQ = *,E20.10,3X,*HZ*
     1 /T5,*WAVE HEIGHT = *,E20.10,3X,*FT*)
      RETURN
      END
```

AD-A039 510 DAVID W TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CE--ETC F/G 13/10
STRUCTURAL SEAWORTHINESS DIGITAL COMPUTER PROGRAM ROSAS (A CONV--ETC(U)
MAY 77 S CHUANG, E A SCHROEDER, S WYBRANIEC

UNCLASSIFIED DTNSRDC-77-0001 NL

2 OF 2
AD
A039510

END
DATE
FILMED
6-77

Table 12 (Continued)

```
      SUBROUTINE DAUX (T, S,D S)
      REAL KAG,MASS
      DIMENSION KAG(21),RINT(21),EI(20),SHP MS(20),DAMPC(20)
     1 ,SCF(9),BUOY SPG(9),ADD MS(9),VRT MS(20)
     2          ,YS( 91),SF(21),GDOT(21),BDM(20),YDOT(20),YH( 9)
     3          ,DYS( 91),DSF(21),DGDOT(21),DBDM(20),DYDOT(20),DYH( 9)
     4          ,HF(20),S( 91),DS(91 )
      COMMON /HBLOK/ KAG,RINT,EI,SHPMS,DAMPC,DXI,NEVAL,SPEED,
     1 BUOY SPG,SCF,ADDMS,VRTMS,IBEG,IEND
      COMMON / TEST / IND
      EQUIVALENCE
     1  (YS,YH),(YS(10),YDOT),(YS(30),GDOT),(YS(51),BDM),(YS(71),SF)
     2 ,(DYS,DYH),(DYS(10),DYDOT),(DYS(30),DGDOT),(DYS(51),DBDM)
     3 ,(DYS(71),DSF)
      DO  20 K = 1,91
   20 YS(K) = S(K)
      NEVAL = NEVAL + 1
      CALL HYD FRC (T,YH,YDOT,HF)
      DO 100 I = 1,20
      DBDM(I) = EI(I)*(GDOT(I+1) - GDOT(I))/DXI
      DYDOT(I) = (HF(I)-DAMPC(I)*YDOT(I)-(SF(I+1)-SF(I))/DXI)/VRT MS(I)
      IF (I .EQ. 1)  GO TO 100
      DSF(I) = (GDOT(I) - (YDOT(I) - YDOT(I-1))/DXI)*KAG(I)
      DGDOT(I) = ((BDM(I) - BDM(I-1))/DXI - SF(I))/RINT(I)
  100 CONTINUE
      DSF(1)  = 0.
      DSF(21) = 0.
      DGDOT( 1) = DGDOT(2)
      DGDOT(21) = DGDOT(20)
      DO 110 I = 1,9
      K = 2*I+2
  110 DYH(I) = .5*(YDOT(K-1) + YDOT(K))
      DO 120 K = 1,91
  120 DS(K) = DYS(K)
      IF (IND .EQ. 0)                      GO TO  6
      WRITE(6,4) NEVAL
    4 FORMAT(1X//1X,30(1H*)/1X,*NEVAL = *,I10//1X,*YH, YDOT, BDM, GDOT,
     1 SF*)
      WRITE (6,5)  YH, YDOT, BDM, GDOT, SF
      WRITE (6,8)
    8 FORMAT(1X,*DYH, DYDOT, DBDM, DGDOT, DSF*)
      WRITE (6,5) DYH,DYDOT,DBDM,DGDOT,DSF
    5 FORMAT(/9E13.6//2(10E13.6/10E13.6//),2(10E13.6/10E13.6/E13.6//))
    6 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE KUTMER(ND,T,H,YO,EPSE,A,HCX,FIRST)
      DIMENSION YO(110),Y1(110),Y2(110),F0(110),F1(110),F2(110)
      DIMENSION EPSE(110),A(110),NEQN(24)
C
C     PASS TO SSMOOTH CONDITION OF TIME INTERVAL - WHOLE TO HALVES
      COMMON /IHVSS/ IHVS
C
      DATA    HCX2 / 1.E200 /
C
C     ND= NUMBER OF EQUATIONS, NO. OF COMPONENTS OF YO
C     T= INDEPENDENT VARIABLE
C     H=INCREMENT FOR WHICH SOLN IS TO BE RETURNED + OR-
C     YO IS THE VECTOR OF DEPENDENT VARIABLES.  ENTER WITH INITIAL
C        VALUES AT T AND RETURNS WITH VALUES AT T+H
C     EPSE=RELATIVE ERROR CRITERION FOR COMPONENTS OF YO .GT. ABS(A)
C     A=ABSOLUTE ERROR CRITERIA FOR COMPONENTS OF YO .LT. ABS(A)
C NOTE-- EPSE AND A MUST BE SPECIFIED FOR EACH COMPONENT OF THE SYSTEM
C     HCX = THE SMALLEST STEP SIZE USED IN THE INTEGRATION
C     DAUX(T,Y,F) RETURNS THE VECTOR F(T,Y)
C     HCX MAY SPECIFY A BEGINNING STEP SIZE = H / 2**K
C     ON RETURN HCX CONTAINS THE SMALLEST STEP SIZE USED
C     SET FIRST = 0.  FOR THE FIRST STEP OR FOR A NEW VALUE OF H
C                0.5 FOR FIRST STEP OR NEW H, AND HCX SPECIFIED
C                1.0 TO CONTINUE INTEGRATION USING THE SAME H
C                1.5 TO CONTINUE WITH THE SAME H WHILE SPECIFYING HCX
C     FIRST = 2. ON RETURN, IF THE ERROR CRITERION COULD NOT
C               BE MET BY REDUCING THE STEP SIZE BY 1/128
C
      IF (ABS(FIRST).LT.1.) IDBL=0
      IFIRST = IFIX(2.*ABS(FIRST))+1
                                        GO TO (15,10,20,5), IFIRST
    5 IF (HC .EQ. 0.)                   GO TO 10
      HCX = AMIN1 (HC,HCX)
   10 HC  = AMIN1 (H,HCX)
      IHVS = 1
      MNSTPS = IFIX(H/HC + .00001)
      IPLOC = MNSTPS
      FIRST = SIGN(1.5,FIRST)
                                        GO TO 25
   15 HC = H
      IHVS = 1
      IPLOC=1
      FIRST = SIGN(1.,FIRST)
   20 MNSTPS = 1
   25 LOC=0
      INEQN = 0
      IF (HC .EQ. 0.)                   GO TO 240
        HCX=HC
   26 IF (2.*HC .EQ. HCX2)              GO TO 30
   27 HC2 = HC/2.
      HC3 = HC/3.
      HC6 = HC/6.
      HC8 = HC/8.
      HC32 = 3.*HC2
      HC23 = 2.*HC3
      HC38 = 3.*HC8
      HCX2 = 2.*HC
   30 CALL DAUX(T,YO,F0)
      DO 40 I=1,ND
   40 Y1(I)  =YO(I)  +    HC3*F0(I)
      CALL DAUX(T+HC3  , Y1,F1)
      DO 50 I=1,ND
   50 Y1(I)  =YO(I)  + HC6*(F0(I) + F1(I) )
      CALL DAUX(T+HC3  , Y1,F1)
      DO 60 I=1,ND
   60 Y1(I)  =YO(I)  +  HC8*F0(I)+   HC38*F1(I)
      CALL DAUX(T+HC2  , Y1,F2)
      DO 70 I=1,ND
   70 Y1(I)  =YO(I  )+  HC2*F0(I)-  HC32*F1(I)+ HCX2*F2(I)
      CALL DAUX( T+HC,Y1,F1)
      DO 80 I=1,ND
```

Table 12 (Continued)

```
   80 Y2(I)=Y0(I) + HC6*(F0(I) + F1(I) ) + HC23*F2(I)
      INC=0
      DO 110 I=1,ND
      ZZZ=ABS(Y2(I))-A(I)
      IF (ZZZ .GE. 0.)                      GO TO 87
      ERROR=ABS(.2*(Y1(I)   -Y2(I)))
      IF(ERROR - A(I)*EPSE(I))              100,100,90
   87 ERROR=ABS(.2-.2*Y1(I)/Y2(I)   )
      IF (ERROR .LE. EPSE(I))               GO TO 100
   90 INEQN = INEQN + 1
      IF(INEQN .LE. 24)        NEQN(INEQN) = I
      IF (IPLOC .LT. 128*MNSTPS)            GO TO 95
      WRITE(6,92)I,T,ERROR
   92 FORMAT( 18H FOR EQUATION NO. I2,27H, THE RELATIVE ERROR AT T = ,
     1      E12.3, 3HIS   , F12.6 )
        FIRST = 2.
                                            GO TO 327
   95 HC=HC/2.
      IPLOC=2 *IPLOC
      LOC=2 *LOC
      IHVS = IHVS + 1
      IF (IDBL .EQ. 1)         WRITE (6,99) I,T,ERROR
   99 FORMAT (70H KUTMER DOUBLED INTERVAL AND IMMEDIATELY HALVED IT DUE
     1TO EQUATION NO. ,I4,7H AT T = 1PE15.6 / 22H THE RELATIVE ERROR IS
     2 1PE15.6)
        HCX=HC
                                            GO TO 27
C     IF ERROR IS LESS THAN 1/100 EPSILON, TRY TO DOUBLE INTERVAL
  100 IF (ERROR*100. .LE. EPSE(I))          GO TO 110
      INC=1
  110 CONTINUE
      IDBL = 0
      T=T+HC
      DO 112 I=1,ND
  112 Y0(I)   =Y2(I)
      LOC=LOC+1
C     IF ANY OF THESE TESTS IS PASSED, THE INTERVAL WILL NOT BE DOUBLED
      IF (LOC .GE. IPLOC)                   GO TO 210
      IF(INC .NE. 0)                        GO TO 210
      IF(MOD(LOC,2) .NE. 0)                 GO TO 210
      IF(IPLOC .LE. MNSTPS)                 GO TO 210
      HC=2.*HC
      LOC=LOC/2
      IPLOC=IPLOC/2
      IHVS = IHVS - 1
      IDBL = 1
  210 IF(IPLOC-LOC)                         26,327,26
  240 IF( H.NE.0.)                          GO TO 15
      WRITE( 6,241)
  241 FORMAT(5X,47H KUTMER ENTERED WITH ZERO INTEGRATION INTERVAL )
  327 IF (INEQN.EQ.0.OR.FIRST.GE.0.) RETURN
      IF(INEQN .LE. 24)                     GO TO 329
      WRITE (6,328) INEQN
  328 FORMAT (90X,15HINTERVAL HALVED,I3,6H TIMES)
      INEQN = 24
  329 WRITE (6,330) (NEQN(I),I = 1,INEQN)
  330 FORMAT ( 90X,32HINTEGRATION INTERVAL WAS HALVED
     1        /90X,32HDUE TO ERROR IN VARIABLE NUMBER   / (89X,8I4))
      RETURN
      END
```

Table 12 (Continued)

```
      SUBROUTINE PLOTIN
C$    DEBUG
C$    STORES(I,TR,TL,LOC)
C$    CALLS(TRYPIN1,TRYPIN2,PLOT,PLOTFIN)
      DIMENSION BDM(20),NAPPLY(20)
      COMMON YS(91)
      COMMON/PBLOK1/NR,TTT(20),R(20),NSBSTN,NAME,CODE,PHONE
      COMMON/PBLOK4/NREPLT,YLU(2,4)
      DIMENSION YR(9,3)
      COMMON/CBLOK/NCALC,NHSTNS,NGROUP,NL(9),NLINV(9),NNLBNL2(90),
     1 ITYPE(18), YCRITA(18), VCRITA(18),IBSTORE(350),IMSTORE(700),
     3 KWORDS(2),NP(2),IFIRST,NAPPLY
      COMMON /PRNT1/T,PITCH,TLEFT
      EQUIVALENCE(YS(51),BDM)
C-----------------------------------------------------------------------
      COMMON /PBLOK3/ PSTORE(1024,6),LOC,INC,TL,TR,TRR
      COMMON /PLTSLM/ RAWSLM,SSMPTS
      DATA MAXLOC /1024/
C-----------------------------------------------------------------------
C**INITIALIZATION
      DATA JUMP/0/
      IF (NR .LE. 0) RETURN
      IF (JUMP .NE. 0) GO TO 1
      CALL CAMRAV(935)
      CALL IDFRMV(NAME,CODE,PHONE)
      JUMP=1
    1 CONTINUE
      I=0
      CALL TRYPIN1
C**DETERMINE LEFT-HAND ENDPOINT OF TIME AXIS
      LOC=0
    5 I=I+1
      IF (I.LE. NR) GO TO 10
      TL=1.E20
      RETURN
   10 IF (R(I) .LE. 0.) GO TO 5
      TR=R(I)
      CALL TRYPIN2
      TL=TTT(I)
C**RIGHT-HAND ENDPOINT OF TIME AXIS
   15 TEMP=TL+R(I)
      TR=AMIN1(TTT(I+1),TEMP)
      TRR=(TEMP-TR)/R(I)
      IF(JUMP .EQ. 1) RETURN
*
C**STORE FRAME"S WORTH OF DATA THEN PLOT
*
      ENTRY PLOT
      IF (NR .LE. 0) RETURN
      IF ( T .LT. TL) RETURN
      IF (T .GT. TR) GO TO 1000
C**STORE ABSCISSA AND ORDINATES
      LOC=LOC+1
      IF (LOC .LE. MAXLOC) GO TO 500
      INC=1
      LOC=MAXLOC
      GO TO 1020
  500 PSTORE(LOC,6) = T
      PSTORE(LOC,1)=PITCH
      PSTORE(LOC,2)=BDM(NSBSTN)
      CALL SEAGEN(YR(1,1),YR(1,2),YR(1,3),CEL,T)
      PSTORE(LOC,3) = YR(4,1)
C
C     PLOT THE RAW SLAM FORCE ON TOP OF PITCH GRAPH
      PSTORE(LOC,4) = RAWSLM
C
C     PLOT THE SMOOTHED SLAM ON TOP OF PITCH GRAPH
      PSTORE(LOC,5) = SSMPTS
      RETURN
```

Table 12 (Continued)

```
C**PREPARE PLOTS
 1000 INC=0
      IF (LOC .EQ. 0) GO TO 2000
 1020 CALL TRYPLOT
 2000 IF (JUMP .NE. 3) GO TO 2010
      CALL PLTND(0)
      IF(NREPLT .EQ. 2)       ENDFILE 8
      RETURN
 2010 CONTINUE
C**RE-INITIALIZE FOR NEXT FRAME
      LOC=0
      JUMP=2
      TL=TL+R(I)
      IF (TL-.99999*TTT(I+1)) 15,5,5
*
C*PLOT LAST FRAME
*
      ENTRY PLOTFIN
      IF (NR .LE. 0) RETURN
      JUMP=3
      IF(NREPLT .EQ. 3)       GO TO 1020
      GO TO 1000
      END
```

Table 12 (Continued)

```
      SUBROUTINE TRYPIN1
CS    DEBUG
CS    STORES(IP1,IP2,DEL,DELA,YYLU,XXL,XXR,MTR)
      COMMON /PBLOK3/ PSTORE(1024,6),LOC,INC,TL,TR,TRR
      COMMON/PBLOK5/NREPLT,YLU(2,4)
      DIMENSION IYC(2,3),DELA(4),II(3),YYLU(2,3),NCH(3),LABEL(5,4)
      DIMENSION XINC(6), FCTNORM(6), IEXP(4)
      DATA XINC/.3,.5,.75,1.,1.5,2./
      DATA LABEL(1,1) / 10HPITCH (DEG    /
      DATA LABEL(1,2) / 31HMIDSHIP BENDING MOMENT (FT-TONS  /
      DATA LABEL(1,3) / 26HWAVE HEIGHT AT MIDSHIP (FT   /
      DATA NCH/10,31,26/
      DATA IYC/60,345,375,660,690,975/
*
C*INITIALIZATION--SET PARAMETERS FOR PLOTTING ROUTINES
*
C--VERTICAL AXES
      IP1=1
      IP2=3
      JUMP=1
      FCTNORM(4) = 30.
      FCTNORM(5) = 15.
      GO TO 15
C--HORIZONTAL (TIME) AXIS
      ENTRY TRYPIN2
      IP1=IP2=4
      YLU(2,4)=TR
      YLU(1,4) = TL
      JUMP=2
C**CALCULATE GRID LINE SPACINGS
 15   DO 110 IP=IP1,IP2
      DEL=YLU(2,IP)-YLU(1,IP)
      IF (DEL .GT. 0.) GO TO 30
      DELA(IP)=DEL
      GO TO 110
 30   IEXP(IP) = IFIX(ALOG10(DEL))
      FNIP = 10. ** IEXP(IP)
      DELNORM = DEL / FNIP
      IF (DELNORM .LT. 10) GO TO 40
      IF=6
      GO TO 90
 40   CONTINUE
      DO 75 IF=1,6
      IEMP=DELNORM/XINC(IF)
      IF (IEMP .EQ. 3 .OR. IEMP .EQ. 4) GO TO 90
 75   CONTINUE
      STOP 11
 90   IF (IP .NE. 4) GO TO 95
      IB=2+MOD(IF,2)
      IC=-IB
      AIB = 2. * FNIP / FLOAT(IB)
      GO TO 100
   95 AIB = FNIP
  100 DELA(IP) = XINC(IF)*AIB
      IF (IP .EQ. 4)  FCTNORM(6) = FNIP
      IF (IP .NE. 4)  FCTNORM(IP) = FNIP
 110  CONTINUE
C  JUMP = 2 IF COMPUTING TIME AXIS DIVISIONS
      IF (JUMP .EQ. 2) GO TO 150
C**DETERMINE WHICH GRID WILL HAVE TIME AXIS LABELING
      K=5
      DO 125 I=1,3
      IEXP(I) = IFIX(ALOG10(AMAX1(YLU(2,I),YLU(1,I))))
      FCTNORM(I) = 10.**IEXP(I)
      IF(IEXP(I).LT.0 .OR. IEXP(I).GT.2)   GO TO 120
      FCTNORM(I) = 1.
      IEXP(I) = 0
  120 DELA(I) = DELA(I)/FCTNORM(I)
      II(I) = 0
      IF (DELA(I) .GT. 0.) K=MIN0(K,I)
      JJ=1
C**MULTIPLY ENDPOINTS OF VERTICAL AXES BY FUDGE FACTORS
      DO 125 J=1,2
      JJ=-JJ
```

Table 12 (Continued)

```
  125 YYLU(J,I) = (1.+JJ*SIGN(.0001,YLU(J,I)))*YLU(J,I)/FCTNORM(I)
      RETURN
C**SET RE-TRACING FREQUENCY FOR VERTICAL GRID LINES
  150 IF (K .NE. 5) II(K)=IC
      RETURN
*
C*  PLOT DATA, 3 PLOTS PER FRAME
*
      ENTRY TRYPLOT
      IF (K .EQ. 5) RETURN
      GO TO (201,202,203),NREPLT
  202 WRITE(8) LOC,TL,TR,TRR
      WRITE (8)  DELA(4),FCTNORM(6),IEXP(4)
      WRITE(8)  ((PSTORE(I,J),J=1,6),I=1,LOC)
      WRITE(6,210)
  210 FORMAT(1X,*PLOT DATA SAVED ON TAPE8*)
      GO TO 201
  203 READ (8) LOC,TL,TR,TRR
      READ (8)  DELA(4),FCTNORM(6),IEXP(4)
      READ(8) ((PSTORE(I,J),J=1,6),I=1,LOC)
      TRR = 1./12.
      DELA(4)=10.
      WRITE(6,1000) LOC,TL,TR,TRR
 1000 FORMAT(I10,4E15.5)
  201 CONTINUE
C**MULTIPLY ENDPOINTS OF HORIZONTAL (TIME) AXIS BY FUDGE FACTOR
      XXL=(1.-SIGN(.0001,TL))*TL
      XXR=(1.+SIGN(.0001,TR))*TR
C**RIGHT-HAND MARGIN
      MTR=TRR*1023
C**DRAW GRID,PLOT,AND ANNOTATE
      L=0
      KK=0
      DO 400 I=1,5
      IF (I .EQ. 4   .OR.  I .EQ. 5)  GO TO 255
      IF (DELA(I) .LE. 0.) GO TO 400
      KK=KK+1
      IF (KK .GT. 1) L=2
      CALL SETMIV(24,MTR,IYC(1,I),1023-IYC(2,I))
      CALL GRID1V(L,XXL,XXR,YYLU(1,I),YYLU(2,I),DELA(4),DELA(I),
     10,0,-1,-1,4,4)
  255 DO 250 J=1,LOC
      IX1 = IXV(PSTORE(J,6))
      IY1 = IYV(PSTORE(J,I) / FCTNORM(I))
      CALL POINTV(IX1,IY1,0)
      IF (J.NE.1) CALL LINEV(IX1,IY1,IX2,IY2)
      IX2 = IX1
  250 IY2 = IY1
      IF (I .EQ. 4   .OR.  I .EQ. 5)  GO TO 400
      IX1 = 12
      IY1 = IYC(2,I)+4
      CALL PRINTV(NCH(I),LABEL(1,I),IX1,IY1)
      IX1 = 12+8*NCH(I)
      IF (IEXP(I).EQ.0) GO TO 270
      CALL PRINTV(8,8H X 10   ),IX1,IY1)
      AEXP = IEXP(I)
      CALL BNBCDV(AEXP,BCD,IX2)
      IX1 = 8*(NCH(I)+7-IX2)
      SGN = 1H
      IF(IEXP(I) .GT. 0)      SGN = 1H-
      IY1 = IY1+8
      CALL PRINTV(1,SGN,IX1,IY1)
      IX1 = IX1+8
      CALL PRINTV(-IX2,BCD,IX1,IY1)
      GO TO 280
  270 CALL PRINTV(1,1H),IX1,IY1)
  280 CONTINUE
      IF (II(I) .NE. 0) CALL PRINTV(-11,11HTIME (SECS),(959-MTR)/2,IYC(1
     1,I)-8)
  400 CONTINUE
      IF (INC .EQ. 1) CALL PRINTV(-55,55HINCOMPLETE PLOT DUE TO PLOT FRE
     1QUENCY EXCEEDING STORAGE  ,30,IYC(1,1)-20)
      RETURN
      END
```

```
      SUBROUTINE MYEND(A,B,C)                                           MYEND
      DIMENSION A(17)                                                   MYEND
10    FORMAT(5(5X,O20))                                                 MYEND
      ENDFILE 6                                                         MYEND
      CALL PLOTFIN                                                      MYEND
      WRITE(6,10) A,C                                                   MYEND
      STOP 6                                                            MYEND
      END                                                               MYEND
```

```
      SUBROUTINE BUILD(I,IA,N,NP)                                       BUILD
C                                                                       BUILD
C  *I* IS A POSSIBLE ADDITION TO ARRAY*IA* OF SIZE *N*.                 BUILD
C  *IA* IS SEARCHED FOR *I*.  IF FOUND, NO ACTION IS TAKEN.             BUILD
C  IF NOT FOUND, *I* IS ADDED TO *IA* SO THAT *IA* IS IN INCREASING ORDE BUILD
C  *N* IS THEN INCREASED BU 1.  *NP* IS THE SMALLEST POWER OF 2 GREATER BUILD
C  THAN OR EQUAL TO N+1.                                                BUILD
      DIMENSION IA(1)                                                   BUILD
      IF (N .EQ. 0) NP=1                                                BUILD
      CALL ISEARCH(I,IA,N,NP,IL,ISTATE)                                 BUILD
      IF  (ISTATE .EQ. 1) RETURN                                        BUILD
      IF  (IL .GT. N) GO TO 7                                           BUILD
      NN=N+1                                                            BUILD
      DO  5  J= IL,N                                                    BUILD
      NN=NN-1                                                           BUILD
5     IA(NN+1)=IA(NN)                                                   BUILD
7     IA(IL)=I                                                          BUILD
      N=N+1                                                             BUILD
      NP=IPWR2(N)                                                       BUILD
      RETURN                                                            BUILD
      END                                                               BUILD
```

Table 12 (Continued)

```
      SUBROUTINE FIELDS(IFORMAT,N,IBUFFER)
      COMMON/FBLOK/FIELD(6) ,IF(6)
      DIMENSION IALPHA(15),LEN(6)
      DIMENSION IBUFFER(1)
      DATA LEN/60,30,20,15,12,10/
      DATA IALPHA/1R0,1R1,1R2,1R3,1R4,1R5,1R6,1R7,1R8,1R9,1R.,1R-,1R+,
     1 1RE,1R /
      ICODE=IFORMAT
      NN=N+1
      I1=61
      DO 100 J=1,N
C$    STORES(IF,FIELD,JJ)
      JJ=NN-J
      ITYPE=ICODE.AND.3B
      ICODE=SHIFT(ICODE,-2)
      I2=I1-1
      I1=I1-LEN(N)
      IF (ITYPE .EQ. 0) GO TO 100
      IBASE1=IBASE2=0
      IQUANT1=IQUANT2=0
      FACTOR=1.
      IDP=0
      IE=0
      SIGN1=1.
      ISIGN2=1
      DO 80 I=I1,I2
      DO 45 K=1,15
      IF (IBUFFER(I) .EQ. IALPHA(K)) GO TO(60,60,60,60,60,60,60,60,60,60
     1,62,64,66,68,80) K
   45 CONTINUE
      GO TO 80
   60 IF (IE .EQ. 1) GO TO 61
      IBASE1=10*IBASE1+K-1
      IQUANT1=IQUANT1+1
      IF (IDP .EQ. 1) FACTOR=FACTOR/10.
      GO TO 80
   61 IBASE2=10*IBASE2+K-1
      IQUANT2=IQUANT2+1
      GO TO 80
   62 IDP=1
      GO TO 80
   64 IF (IE .EQ. 1 .OR. IQUANT1 .NE. 0) GO TO 65
      SIGN1=-SIGN1
      GO TO 80
   65 IF (IE .NE. 1) IE=1
      ISIGN2=-ISIGN2
      GO TO 80
   66 IF (IQUANT1 .EQ. 0) GO TO 80
   68 IE=1
   80 CONTINUE
      IF (ITYPE .EQ. 2) GO TO 90
      ISIGN1=SIGN1
      IFACTOR=FACTOR
      IF (JJ)=ISIGN1*IBASE1*IFACTOR
      GO TO 100
   90 IF (IQUANT2 .EQ. 0) GO TO 94
      FACTOR2=10.**(ISIGN2*IBASE2)
      GO TO 96
   94 FACTOR2=1.
   96 FIELD(JJ)=SIGN1*FACTOR*FLOAT(IBASE1)*FACTOR2
C$    OFF(STORES)
  100 CONTINUE
      RETURN
      END
```

Table 12 (Continued)

```
      SUBROUTINE HSTDFN(IBUFFER,NTYPMX,NTYPES,BIAS,WIDTH,XMAX)          HST
      DIMENSION BIAS(NTYPMX),IBUFFER(60),WIDTH(NTYPMX),IALPHA(17)       HST
      DATA IALPHA /1R0,1R1,1R2,1R3,1R4,1R5,1R6,1R7,1R8,1R9,1R.,1R-,1R(, HST
     1 1R),1R  ,1R,,1R+/                                                HST
      NTYPES=0                                                          HST
      IEND=0                                                            HST
      ISTART=1                                                          HST
      IOPT=0                                                            HST
5     IBASE=0                                                           HST
      IQUANT=0                                                          HST
      FACTOR=1.                                                         HST
      IDP=0                                                             HST
      SIGN=1.                                                           HST
      IOPT=IOPT+1                                                       HST
      IF (IOPT .EQ. 3) IOPT=1                                           HST
      IF(ISTART .EQ. 0) GO TO 450                                       HST
      ISTART=0                                                          HST
      DO 450 L=1,60                                                     HST
      DO 400 M=1,17                                                     HST
      IF (IBUFFER(L) .EQ. IALPHA(M)) GO TO (410,410,410,410,410,410,    HST
     1 410,410,410,410,420,430,440,440,440,450,450     )M               HST
400   CONTINUE                                                          HST
      GO TO 900                                                         HST
410   IQUANT=IQUANT+1                                                   HST
      IBASE=10*IBASE+M-1                                                HST
      IF (IDP .EQ. 1) FACTOR=FACTOR/10.                                 HST
      GO TO 450                                                         HST
420   IDP=1                                                             HST
      GO TO 450                                                         HST
430   SIGN=-SIGN                                                        HST
      GO TO 450                                                         HST
440   IF (IQUANT .NE. 0) GO TO 910                                      HST
450   CONTINUE                                                          HST
500   CONTINUE                                                          HST
900   IEND=1                                                            HST
      IF (IQUANT .EQ. 0) GO TO 1900                                     HST
910   IF (IOPT .EQ. 2) GO TO 1000                                       HST
      IF (NTYPES .EQ. 0) GO TO 920                                      HST
      BIAS(NTYPES)=XMAX                                                 HST
920   XMAX=SIGN*FACTOR*FLOAT(IBASE)                                     HST
      IF (IEND .EQ. 1) GO TO 1900                                       HST
      GO TO 5                                                           HST
1000  IF (IEND .EQ. 1) GO TO 1900                                       HST
      NTYPES=NTYPES+1                                                   HST
      IF (NTYPES .GT. NTYPMX) GO TO 1890                                HST
      WIDTH(NTYPES)=SIGN*FACT)R*FLOAT(IBASE)                            HST
      GO TO 5                                                           HST
1890  NTYPES=NTYPES-1                                                   HST
1900  RETURN                                                            HST
      END                                                               HST
```

Table 12 (Continued)

```
      SUBROUTINE INUPH
C$    DEBUG
C$    ARRAYS
C** INPUT DATA HANDLER
*
C**READS INPUT CARDS AND PERFORMS LIMITED INITIALIZATION.  INPUT
C**FEATURES ARE THE FOLLOWING--
C    1.  COLS 1-10 ARE TH  MNEMONIC FIELD, COLS 71-80 THE CONTINUATION
C  FIELD, AND COLS 11-70 THE DATA FIELDS (SIX 10 COLUMN FIELDS OR THREE
C  20 COLUMN FIELDS)
C    2. A CONTINUATION CARD HAS A + IN COL 1 FOLLOWED BY THE NON-INITIAL
C  CHARACTERS OF A CONTINUATION FIELD.
C    3.  PARAMETERS FOR WHICH NO INPUT CARDS APPEAR WILL MAINTAIN THE
C  VALUES THEY  HAD FOR THE PREVIOUS CASE.  FOR THE FIRST CASE, SUCH
C  PARAMETERS WILL BE SET TO DEFAULT VALUES.
C    4.  INPUT CARDS MAY APPEAR IN ANY ORDER.
C    5.  A SINGLE END OF FILE REPRESENTS END-OF-CASE.  A DOUBLE END OF FI
C  END-OF JOB.
C
      INTEGER CNTNATN
      COMMON /INPT1/IEOF,B(11,2),TSTRT,NHALVES,TF,W(10,2),NW(2),EPS(91),
     1ABSERR(91),XLNGTH ,TCPU,CONTROL(4)
      COMMON/INPT2/J,ISFIELD,BUFFER(60),ICFIELD
C-----------------------------------------------------------------------
      DIMENSION ICNTAR(3,100),MNEREF(22)
      DATA ISTCNT,NMNES/100,22/
C-----------------------------------------------------------------------
      DATA MNEREF/4LSTAT,4LHALF,4LINIT,4LHYDR,4LNLAD,4LNLBU,4LNL1A,
     1                                                          4LNL1B,
     14LNL2A,4LNL2B,4LINTG,4LDEBU,4LSEA-,4LSPEE,4LCONT,4LPRIN,4LCOMM,
     2 4LIDEN,  4LSCTN,4LPLOT,4LAXES,4LDAMP/
      DATA IFIRST /0/
      IF (IFIRST .EQ. 1) GO TO 1
      IFIRST=1
C**JOB INITIALIZATION
      IEOF=0
C**SET DEFAULT VALUES
      CALL NLIN
      CALL STIN(0)
      CALL HSTIN(0)
      CALL MSCON(0)
      CALL HYDIN(0)
      CALL CNTRN
      CALL INPLTIN
      CALL INPRIN
      CALL INSEGIN(NULL,NULL,NULL,NULL,NULL)
      CALL INTRIN(0)
C**CASE INITIALIZATION
C     PATCH FOR SLAM ENTRIES
      CALL INSMIN(NULL)
      CALL INSMOT(NULL)
C
 1    ICARDS =0
      CALL INTIN
      CALL INDMPIN
      ITIMES=0
      ICFIELD=0
      ISTORE=JSTORE=1
      NSCNTN=0
      IS=0
      NTAPE=5
      N1=7
      WRITE(6,2)
 2    FORMAT(1H1,*INPUT CARDS*)
C* READ A CARD
 5    READ (NTAPE, 10) MFIELD, BUFFER, CNTNATN
 10   FORMAT(A10,60R1,A10)
      IF (EOF(NTAPE)) 30,50
C**CHECK FOR END-OF-JOB
 30   IF (ICARDS .GT. 0) GO TO 40
      IEOF=1
      RETURN
```

```
C**CHECK SCRATCH DISK FOR CONTINUATION CARDS
 40   IF (IS .NE. 0) GO TO 45
C**COMPLETE INPUT OPERATIONS AND RETURN
 42   CALL NLFIN
      CALL STFIN(NULL)
      CALL HSTFIN(NULL)
      CALL INTRFIN(NULL)
      CALL MSFIN(NULL)
      CALL HYDFIN(NULL)
      CALL CNTFIN
      CALL INPLFIN
      CALL INPRFIN
      CALL INSGFIN(NULL,NULL,NULL,NULL,NULL)
      CALL INSMFIN(NULL)
      RETURN
C**PROCESS CONTINUATION CARDS ON SCRATCH DISK
 45   ITIMES=ITIMES+1
      IF (ITIMES .LT. 5) GO TO 47
      WRITE(6,46)
 46   FORMAT(1X,*CONTINUATION FIELD ERROR(S)*)
      GO TO 42
 47   IF (NTAPE .NE. 5) REWIND NTAPE
      REWIND N1
      NTAPE=N1
      N1=15-NTAPE
      IS=0
      GO TO 5
C**CHECK COLUMN 1 FOR +
 50   ICARDS=ICARDS+1
      ICOL1=MFIELD .AND. 77000000000000000000B
      IF (ICOL1 .NE. 1L+) GO TO 500
C**CHECK IF CARD PRECEEDING CONTINUATION CARD HAS BEEN READ
      IF (NSCNTN .EQ. 0) GO TO 400
C$    STORES(IS,ICOL2X)
      ICOL2X=SHIFT(MFIELD-ICOL1,6)
      DO 100 I=1,NSCNTN
C$    GOTOS
      IF (ICNTAR(1,I) .EQ. ICOL2X) GO TO 450
 100  CONTINUE
C**WRITE CONTINUATION CARD ON SCRATCH DISK
 400  IS=IS+1
C$    OFF(GOTOS)
C$    OFF(STORES)
      WRITE(N1,10) MFIELD, BUFFER , CNTNATN
      GO TO 5
C**RETRIEVE MNEMONIC AND SUPPORT FIELD FROM STORAGE ARRAY
 450  MNEMNC=ICNTAR(2,I)
      ISFIELD=ICNTAR(3,I)
C**SET POINTERS INDICATING LOCATIONS AVAILABLE FOR STORAGE PURPOSES
      ICNTAR(2,I)=ISTORE
      ICNTAR(1,I)=10H
C$    STORES(ISTORE)
      IF (I .EQ. NSCNTN) NSCNTN=NSCNTN-1
      ISTORE=I
C$    OFF(STORES)
      GO TO 510
 500  MNEMNC=MFIELD .AND. 77777777000000000000B
      ISFIELD=0
C**SEARCH VOCABULARY FOR MNEMONIC
 510  DO 530 K=1,NMNES
      J=K
      IF (MNEMNC .EQ. MNEREF(K)) GO TO 550
 530  CONTINUE
      WRITE(6,540) MFIELD, BUFFER, CNTNATN
 540  FORMAT(1X/1X,*UNRECOGNIZABLE MNEMONIC*/1X,A10,60R1,A10/)
      GO TO 5
 550  WRITE(6,560) MFIELD,BUFFER,CNTNATN
 560  FORMAT(1X,A10,60R1,A10)
C**MOVE DATA FIELDS INTO APPROPRIATE PARAMETER VARIABLES
      GO TO (600,610,660,670,640,640,640,640,640,640    ,720,700,710,
     1 650,680,700,5,5,650,690,690,730) J
 600  CALL STNDT(NULL) $ GO TO 5
```

Table 12 (Continued)

```
610  CALL HSDT(NULL) $ GO TO 5                                      INPH
640  CALL NLINPUT $ GO TO 1990                                       INPH
650  CALL MSCDT(NULL) $ GO TO 5                                      INPH
660  CALL INITDT $ GO TO 5                                           INPH
670  CALL HYDRDT(NULL) $ GO TO 5                                     INPH
680  CALL CNTRDT $ GO TO 5                                           INPH
690  CALL INPLTDT $ GO TO 5                                          INPH
700  CALL INPRDT $ GO TO 5                                           INPH
710  CALL INSEGDT(NULL,NULL,NULL,NULL,NULL) $ GO TO 5                INPH
 720  CALL INTRDT(NULL) $GO TO 1990                                  INPH
 730  CALL INDMPDT $ GO TO 5                                         INPH
C**CHECK IF CONTINATION FIELD EXPECTED                               INPH
1990  IF (ICFIELD .EQ. 0) GO TO 5                                    INPH
C**CHECK FOR BLANK CONTINUATION FIELD                                INPH
      IF (CNTNATN .EQ. 10H           ) GO TO 5                       INPH
C**STORE CONTINUATION FIELD                                          INPH
      ICOL2X=CNTNATN                                                 INPH
      DO 2000 I=1,10                                                 INPH
      IF((ICOL2X .AND. 77000000000000000000B ).NE. 1L   ) GO TO 2010 INPH
 2000 ICOL2X=SHIFT(ICOL2X,6)                                         INPH
C$    STORES(ICNTAR)                                                 INPH
 2010 ICNTAR(1,ISTORE)=SHIFT(ICOL2X.AND. 00777777777777777777B,6)    INPH
C**STORE MNEMONIC FIELD AND A SUPPORT FIELD                          INPH
      KSTORE=ICNTAR(2,ISTORE)                                        INPH
      IF (ISTORE .LE. ISTCNT) GO TO 2020                             INPH
      WRITE(6,2015)                                                  INPH
 2015 FORMAT(1X,*DECK SCRAMBLING EXCEEDS STORAGE CAPABILITIES.*/1X,*ORDE INPH
     1R DECK  BETTER OR INCREASE ISTCNT AND SECOND DIMENSION OF ICNTAR.* INPH
     2)                                                              INPH
      STOP                                                           INPH
 2020 CONTINUE                                                       INPH
      ICNTAR(2,ISTORE)=MNEMNC                                        INPH
      ICNTAR(3,ISTORE)=ISFIELD                                       INPH
C$    OFF(STORES)                                                    INPH
C**SET POINTER INDICATING NEXT AVAILABLE STORAGE LOCATION            INPH
      IF (ISTORE .NE. JSTORE) GO TO 2050                             INPH
      KSTORE=JSTORE=JSTORE+1                                         INPH
C$    STORES(ISTORE)                                                 INPH
 2050 IF (ISTORE .GT. NSCNTN) NSCNTN=ISTORE                          INPH
      ISTORE=KSTORE                                                  INPH
C$    OFF(STORES)                                                    INPH
      ICFIELD=1                                                      INPH
      GO TO 5                                                        INPH
      END                                                            INPH


      BLOCK DATA                                                     INPH
      DIMENSION NAPPLY(20)                                           INPH
      COMMON/CBLOK/NCALC,NHSTNS,NGROUP,NL(9),NLINV(9),MNLBNL2(90),   INPH
     1 ITYPE(18), YCRITA(18), VCRITA(18),IBSTORE(350),IMSTORE(700),  INPH
     3 KWORDS(2),NP(2),IFIRST,NAPPLY                                 INPH
      DATA NCALC,NHSTNS/9,20/                                        INPH
      DATA NLINV/3,5,7,9,11,13,15,17,19/                             INPH
      DATA NAPPLY/0,1,1,1,2,2,3,3,4,4,5,5,6,6,7,7,8,8,9,9/           INPH
      END                                                            INPH
```

```
      SUBROUTINE NLINPUT
      COMPLEX GROUP,GROUPA
      COMMON/FBLOK/FIELDA(6) ,IF(6)
      EQUIVALENCE(IF(1),ISTN),(IF(1),ICODE),(IF(2),NEXP),
     1 (FIELDA(2),YC),(FIELDA(3),C(1)),(FIELDA(3),VC),(LOCC(1) ,LOCA),
     2(LOCC(2),LOCB),(AMSTORE(1),IMSTORE(1)),(BSTORE(1),IBSTORE(1))
      COMMON /INPT2/J,ISFIELD,BUFFER(60),ICFIELD
      DIMENSION KWORDS(2) ,NP(2)
      DIMENSION KEWDARY(4,2),NUKEWD(2),NWORDS(2)
      DIMENSION IFRMT(4),C(4),IWORD(5),LOCC(2),
     1 AMSTORE(350,2),BSTORE(350),MC(28),LCL(19),IBSTORE(350)
     1,IMSTORE(350,2)
      COMMON/CBLOK /NCALC,NHSTNS,NGROUP,NL,NLINV,MNLBNL2,ITYPE,YCRITA
     2, VCRITA,IBSTORE,IMSTORE,KWORDS,NP,IFIRST,NAPPLY
C-----------------------------------------------------------------------
      DIMENSION NAPPLY(20)
      DIMENSION NCFS(18),NPEXP(18)
      DIMENSION GROUPA(18),NL(9),NLINV(9),MNLBNL1(5,9,2),ITYPE(18),
     1 YCRITA(18),VCRITA(18),MNLBNL2(5,9,2)
      DATA NPNCMX/18/
      DATA NCALCMX /9/
C-----------------------------------------------------------------------
      DATA IFRMT/2652B,3000B,3200B,3652B/
      DATA NCFS/3*1,2,2*1,2,3,2*1,2,3,4,2*2,3,2,3/
      DATA KEWDARY/2RLL,2RGL,2RLG,2RGG,1RL,1RG/
      DATA NPEXP/5*1,3,2*4,5,6,7,2*8,2,2*14,5,7/
      DATA IFIRST/0/
      DATA MC/0,3*1,3*2,1,2,1,3*3,2,3,2,1,3,2,1,2,3,1,3,1,2*3,2/
      DATA LCL/0,1,2,3,4,5,6,8,10,11,12,14,17,20,21,23,25,26,28/
      DATA NUKEWD/2,4/
      IF (ISFIELD .EQ. 0) GO TO 20
C**UNPACK THE SUPPORT FIELD
      ICARD=MOD(ISFIELD,10000)
      MJSTN=ISFIELD-ICARD
      GO TO 25
C**SET FIRST-CARD INDICCATOR
 20   ICARD=1
C**CATEGORY INDICATORS
C$    STORES(JJ)
 25   JJ=J-4
C$    OFF(STORES)
      MB=2-MOD(JJ,2)
      IT=(JJ+1)/2
      IF (ICARD .NE. 1) GO TO 5000
C**FIRST CARD
      CALL FIELDS(  IFRMT(IT),6,BUFFER( 1))
      VVC=VC
      YYC=YC
      IF (IT .NE. 3)VVC=1.E20
      IF (IT .EQ. 1)YYC=1.E20
C**CHECK IF NEW GROUP
      GROUP=CMPLX(YYC,VVC)
      IF (NGROUP .EQ. 0) GO TO 2320
      DO 2310 I=1,NGROUP
      IF (GROUP .EQ. GROUPA(I)) GO TO 2325
 2310  CONTINUE
 2320  NGROUP=NGROUP+1
      I=NGROUP
      GROUPA(NGROUP)=GROUP
      YCRITA(NGROUP)=YYC
      VCRITA(NGROUP)=VVC
      ITYPE(NGROUP)=IT+   ((IT-1)/2)
C**PREPARE GROUP CODE WORD
 2325 MJSTN=10000*ISTN
      IWORD(1)=MJSTN+I
C**PREPARE FILLER CODE WORD(S)
      II=4-IT
      INSERT=II+II/2
      IF (INSERT .EQ. 1) GO TO 2400
      ITEMP=MJSTN+9900
      DO 2350 I=2,INSERT
 2350 IWORD(I)=ITEMP+I
```

Table 12 (Continued)

```
C**INSERT WORD(S)
      IKEY=0
 2400 DO 2420 I=1,INSERT
 2420 CALL BUILD(IWORD(I),MNLBNL1(1,1,MB),NWORDS(MB),NP)
      IF (IT .NE. 1 .OR. ICARD .NE. 1) GO TO 8000
      IF (IT .EQ. 1 .AND. IKEY .EQ. 1) GO TO 8000
      IKEY=1
      GO TO 5010
C**SUBSEQUENT CARDS
 5000 CALL FIELDS(IFRMT(4),6,BUFFER( 1))
 5010 LOC=LOCC(MB)=LOCC(MB)+1
      IF (NEXP .GE. 1 .AND. NEXP .LE. NPNCMX) GO TO 5015
      WRITE(6,5012)
 5012 FORMAT(1X,*ERROR IN COLS 21 - 30.  CARD IGNORED.*/)
      GO TO 8000
 5015 CONTINUE
C$    STORES(NN)
      NN=NCFS(NEXP)
C$    OFF(STORES)
      IF (MB .EQ. 2) GO TO 6000
C**ADDED MASS COEFICIENTS AND FIRST DERIVATIVE
      IMSTORE(LOC,1)=NEXP
      IMSTORE(LOC,2)=NPEXP(NEXP)
      DO 5345 I=1,NN
      LOCA=LOCA+1
      AMSTORE(LOCA,1)=C(I)
 5345 AMSTORE(LOCA,2)=MC(LCL(NEXP)+I)*C(I)
      GO TO 7000
C**BUOYANCY COEFICIENTS
 6000 IBSTORE(LOCB)=NEXP
      DO 6350 I=1,NN
      LOCB=LOCB+1
 6350 BSTORE(LOCB)=C(I)
C**DECIPHER CONDITION CODE
 7000 IF (IT .EQ. 1) GO TO 7500
      CALL SCAN(BUFFER, 1,10,KEWDARY(1,4-IT),NUKEWD(IT-1),IKEY,NULL)
      IF (IKEY .NE. 0) GO TO 7500
      WRITE(6,7100)
 7100 FORMAT(1X,*ILLEGAL CONDITION CODE.  CARD IGNORED.*/)
      GO TO 8000
C**PREPARE COEFICIENT/LOCATION CODE WORD
 7500 IWORD(1)=MJSTN+1000*IKEY+LOC
      INSERT=1
      GO TO 2400
C**SET INDICATORS FOR SUBSEQUENT CARDS
 8000 IF (ICARD .EQ. 2*IT-1) GO TO 9000
      ICFIELD=1
      ISFIELD=MJSTN+ICARD +1
      RETURN
 9000 ICFIELD=0
      RETURN
C**DECODE ARRAY OF CODE WORDS
      ENTRY NLFIN
      IF (NWORDS(1) .EQ. KWORDS(1) .AND. NWORDS(2) .EQ. KWORDS(2))
     1 RETURN
      IF (NWORDS(1)/5*5.NE. NWORDS(1) .OR. NWORDS(2)/5*5 .NE. NWORDS(2))
     1 GO TO 10100
      KWORDS(1)=NWORDS(1)
      KWORDS(2)=NWORDS(2)
      NWORDS(1)=NWORDS(1)/5
      NWORDS(2)=NWORDS(2)/5
      DO 9180 IK=1,2
      IW=NWORDS(IK)
      DO 9180 IJ=1,IW
      KSTN=MNLBNL1(1,IJ,IK)/10000
      DO 9120 IJA=1,NCALC
      IF (KSTN .EQ. NLINV(IJA)) GO TO 9130
 9120 CONTINUE
      GO TO 13000
 9130 NL(IJA)=NL(IJA)+3-IK
      DO 9180 IJA=1,5
```

Table 12 (Continued)

```
 9180 MNLBNL2(IJA,IJ,IK)=MOD(MNLBNL1(IJA,IJ,IK),1000)
      NWORDS(1)=KWORDS(1)
      NWORDS(2)=KWORDS(2)
      RETURN
C**ERROR MESSAGES
10100 WRITE (6,10200)
10200 FORMAT(1X,*COUNT ERROR IN NL-- CARDS*)
      STOP
13000 WRITE(6,13100)  KSTN
13100 FORMAT(1X,*NL-- CARDS AND HYDRO-DATA DO NOT AGREE AT STATION*,
     1 I5)
      STOP
C**INITIALIZATION
      ENTRY NLIN
      IF (IFIRST .EQ. 0) KWORDS(1)=KWORDS(2)=0
      NWORDS(1)=KWORDS(1)
      NWORDS(2)=KWORDS(2)
      IF (IFIRST .EQ. 1) RETURN
      IFIRST=1
      NGROUP=0
      LOCA=LOCB=0
      NWORDS(1)=NWORDS(2)=0
      DO 1400 I =1,NCALCMX
 1400 NL(I)=0
      RETURN
      END
```

Table 12 (Continued)

```
      SUBROUTINE CNTRN                                                  INCN     2
C                                                                       INCN     3
C** CONTROL PARAMETERS                                                  INCN     4
C                                                                       INCN     5
      DIMENSION KENDARY(1)                                              INCN     6
      COMMON/INPT2/J,ISFIELD,IBUFFER(60),ICFIELD                        INCN     7
      COMMON/INPT1/IEOF,B(11,2),TSTRT,HMNI,TF,W(10,2),NW(2),EPS(91) ,   CO5      2
     1ABSERR(91),XLNGTH ,TCPU,CONTROL(4)                                CO5      3
C**SET DEFAULTS                                                         INCN     9
      CONTROL(2)=CONTROL(3)=CONTROL(4)=2HNO                             INCN    10
      RETURN                                                            INCN    11
C**ACCEPT DATA                                                          INCN    12
      ENTRY CNTRDT                                                      INCN    13
      DO 10 K=1,3                                                       INCN    14
      LS1=20*K-9                                                        INCN    15
      LS2=LS1+9                                                         INCN    16
      DO 5 KK =LS1,LS2                                                  INCN    17
      IF (IBUFFER(KK) .EQ. 1RY) GO TO 7                                 INCN    18
 5    CONTINUE                                                          INCN    19
      GO TO 10                                                          INCN    20
 7    CONTROL(K+1)=3HYES                                                INCN    21
10    CONTINUE                                                          INCN    22
      RETURN                                                            INCN    23
C**DATA ECHO                                                            INCN    24
      ENTRY CNTFIN                                                      INCN    25
      WRITE (6,50) (CONTROL(L),L=2,4)                                   INCN    26
 50   FORMAT(1H-,*CONT<OL OPTIONS*/T10,*PUNCH FINAL CONDITIONS*,T40,A3/ INCN    27
     1 T10,*NON-LINEAR HYDRO FORCE*,T40,A3/T10,*HYDRO DAMPING*,T40,A3)  INCN    28
      RETURN                                                            INCN    29
      END                                                               INCN    30
```

Table 12 (Continued)

```
      SUBROUTINE HSTIN (ITS)
C
C**HALF STATION DATA
C
      COMMON /HBLOK/ KAG(21),RINT(21),EI(20),SHPMS(20),DAMPC(20),
     1 DXI,NEVAL,SPEED,BUOY SPG(9),SCF(9),ADDMS(9),VRTMS(20),IBEG,IEND
      COMMON /INPT2/J,ISFIELD,BUFFER(60),ICFIELD
      DIMENSION YS(91),SF(21),GDOT(21),BDM(20),YDOT(20),YH(9)
      COMMON/FBLOK/FIELDA(6) ,IF(6)
      COMMON //YS
      EQUIVALENCE
     1  (YS,YH),(YS(10),YDOT),(YS(30),GDOT),(YS(51),BDM),(YS(71),SF)
      DATA KX/0/
      IF (ITS .EQ. 0) GO TO 4
      GO TO (2,3) ITS
 2    ITC =1
      RETURN
 3    IDC=1
      RETURN
C**SET DEFAULT VALUES
  4   ITC=IDC=0
      IF (KX .EQ. 1) RETURN
      KX=1
      DO 5 I=1,20
 5    SHPMS(I)=EI(I)=DAMPC(I)=YDOT(I)=BDM(I)=0.
      RETURN
C**ACCEPT DATA
      ENTRY HSDT
      CALL FIELDS(52520,6,BUFFER)
      I=FIELDA(1)+.500001
      SHPMS(I)=FIELDA(2)
      EI(I)=FIELDA(3)
      IF (IDC .EQ. 1) GO TO 10
      DAMPC(I)=FIELDA(4)
 10   IF (ITC .EQ. 1) RETURN
      YDOT(I)=FIELDA(5)
      BDM(I)=FIELDA(6)
      RETURN
C**DATA ECHO
      ENTRY HSTFIN
      CALL INDMFIN
      WRITE(6,20)
 20   FORMAT(1H-,*HALF STATION DATA*///T10,*SHPMS = MASS PER UNIT LENGTH
     1*/T10,                                  *EI = BENDING RIGIDITY
     1*/T10,*DAMPC = STRUCTURAL DAMPING COEFICIENT*/T10,*YDOT = INITIAL
     2 VERTICAL VELOCITY*/T10,*BDM = INITIAL BENDING MOMENT*///
     3 1X,*HALF-STATION*,T19,* SHPMS*,T34,*EI*,T49,*DAMPC*,T64,*YDOT*,
     4 T79,*BDM*)
      DO 40 L=1,20
      H=L-.5
 40   WRITE(6,50) H,SHPMS(L),EI(L),DAMPC(L),YDOT(L),BDM(L)
 50   FORMAT(1X,F5.1,T16,5E15.7)
      END
```

Table 12 (Continued)

```
      SUBROUTINE HYDIN(ITS)                                              INHYD    2
C                                                                        INHYD    3
C** HYDRO DATA                                                           INHYD    4
C                                                                        INHYD    5
      COMMON /INPT2/J,ISFIELD,BUFFER(60),ICFIELD                         INHYD    6
      COMMON /HDBLOK/ KAG(21),RINT(21),EI(20),SHPMS(20),DAMPC(20),       CO1      2
     1 DXI,NEVAL,SPEED,BUOY SPG(9),SCF(9),ADDMS(9),VRTMS(20),IBEG,IEND   CO1      3
      COMMON/FBLOK/FIELDA(6) ,IF(6)                                      CO2      2
      DIMENSION YS(91),SF(21),GDOT(21),BDM(20),YDOT(20),YH(9)            INHYD    9
      COMMON //YS                                                        INHYD   10
      EQUIVALENCE                                                        CO3      2
     1  (YS,YH),(YS(10),YDOT),(YS(30),GDOT),(YS(51),BDM),(YS(71),SF)     CO3      3
      COMMON /HDBLOK/HYDDAMP(9)                                          INHYD   12
      DATA KX /0/                                                        INHYD   13
      IF (ITS .EQ. 0) GO TO 1                                            INHYD   14
      ITC = 1                                                            INHYD   15
      RETURN                                                             INHYD   16
C**SET DEFAULT VALUES                                                    INHYD   17
 1    ITC=1                                                              INHYD   18
      IF (KX .EQ. 1) RETURN                                              INHYD   19
      KX=1                                                               INHYD   20
      DO 5 I=1,9                                                         INHYD   21
 5    ADDMS(I)=BUOY SPG(I)=SCF(I)=HYDDAMP(I)=YH(I)=0.                    INHYD   22
      RETURN                                                             INHYD   23
C**ACCEPT DATA                                                           INHYD   24
      ENTRY HYDRDT                                                       INHYD   25
      CALL FIELDS(32528,6,BUFFER)                                        INHYD   26
      IF (IF(1) .LT. 3 .OR. IF(1) .GT. 19 .OR. (IF(1)/2)*2 .EQ. IF(1))   INHYD   27
     1 GO TO 90                                                          INHYD   28
      I= (IF(1)-1)/2                                                     INHYD   29
      ADDMS(I)=FIELDA(2)                                                 INHYD   30
      BUOY SPG(I)=FIELDA(3)                                              INHYD   31
      SCF(I)=FIELDA(4)                                                   INHYD   32
      HYDDAMP(I)=FIELDA(5)                                               INHYD   33
      IF (ITC .EQ. 1) RETURN                                             INHYD   34
      YH(I)=FIELDA(6)                                                    INHYD   35
      RETURN                                                             INHYD   36
      ENTRY HERR                                                         INHYD   37
 90   WRITE (6,100) IF (1)                                               INHYD   38
 100  FORMAT(1X,*HYDRO-FORCE CANNOT BE CALCULATED AT STATION*,I10)       INHYD   39
      RETURN                                                             INHYD   40
C**DATA ECHO                                                             INHYD   41
      ENTRY HYDFIN                                                       INHYD   42
      WRITE(6,200)                                                       INHYD   43
 200  FORMAT(1H-,*HYDRO-FORCE DATA*//T10,*ADDMS = LINEAR ADDED MASS*/    INHYD   44
     1 T10,*BUOY = LINEAR BYOYANCY*/T10,*SCF = SMITH CORRECTION*/,T10,*H INHYD   45
     2DMP = HYDRO-DAMPING*/T10,*YH = INITIAL VERTICAL DISPLACEMENT*///   INHYD   46
     3 1X,*STATION*,T20,*ADDMS*,T35,*BUOY*,T50,*SCF*,T65,*HDMP*,T80,     INHYD   47
     4*YH*)                                                              INHYD   48
      DO 250 L=1,9                                                       INHYD   49
      I=2*L+1                                                            INHYD   50
250   WRITE(6,300) I,ADDMS(L),BUOYSPG(L),SCF(L),HYDDAMP(L),YH(L)         INHYD   51
 300  FORMAT(1X,I5,10X,5E15.7)                                           INHYD   52
      END                                                                INHYD   53
```

Table 12 (Continued)

```
      SUBROUTINE INDMPIN                                                DMP      2
C                                                                       DMP      3
C** PARAMETERS FOR CALCULATING STRUCTURAL DAMPINF                       DMP      4
C                                                                       DMP      5
      COMMON /HBLOK/ KAG(21),RINT(21),EI(20),SHPMS(20),DAMPC(20),      C01      2
     1 DXI,NEVAL,SPEED,BUOY SPG(9),SCF(9),ADDMS(9),VRTMS(20),IBEG,IEND  C01      3
      COMMON/FBLOK/FIELDA(6) ,IF(6)                                     C02      2
      COMMON/INPT2/J,ISFIELD,BUFFER(60),ICFIELD                         DMP      8
      DATA KX,PI/0, 3.1415926535898/                                    DMP      9
C**ALLOW COEFICIENTS TO BE ON HALF-STATION CARDS                         DMP     10
      CALL HSTIN(0)                                                     DMP     11
      FPI = 2.                                                          DMP     12
      RETURN                                                            DMP     13
C**DAMPC CARD DETECTED                                                  DMP     14
      ENTRY INDMPDT                                                     DMP     15
      KX=1                                                              DMP     16
      CALL HSTIN(2)                                                     DMP     17
      CALL FIELDS(5000B,6,BUFFER)                                       DMP     18
      CPSNAT=FIELDA(1)                                                  DMP     19
      PCCD=FIELDA(2)                                                    DMP     20
      RETURN                                                            DMP     21
C**CALCULATE DAMPING COEFICIENT                                         DMP     22
      ENTRY INDMFIN                                                     DMP     23
      CALL HFCIN(NULL,NULL,NULL,NULL)                                   DMP     24
      IF (KX .EQ. 0) RETURN                                             DMP     25
      PW=FPI*CPSNAT*PCCD                                                DMP     26
      WRITE (6,1) CPSNAT,PCCD                                           DMP     27
 1    FORMAT(1H-,*STRUCTURAL DAMPING PARAMETERS*/T10,*NATURAL FREQUENCY* DMP     28
     1 ,T30,E20.10,3X,*HZ*/T10,*PERCENT DAMPING*,T30,E20.10)           DMP     29
      DO 5 IH=1,20                                                      DMP     30
 5    DAMPC(IH)=VRTMS(IH)*PW                                            DMP     31
      RETURN                                                            DMP     32
      END                                                               DMP     33
```

Table 12 (Continued)

```
      SUBROUTINE INPLTIN                                                INPL
C                                                                       INPL
C** PLOTING PARAMETERS                                                  INPL
C                                                                       INPL
      DIMENSION DFYLU(3)                                                INPL
      COMMON /INPT2/J,ISFIELD,BUFFER(60),ICFIELD                        INPL
      COMMON/FBLOK/FIELDA(6) ,IF(6)                                     CD2
      COMMON/PBLOK1/NR,TTT(20),R(20),MSBSTN,NAME,CODE,PHONE             CD6
      COMMON/PBLOK4/NREPLT,YLU(2,4)                                     CD7
      COMMON/INPT1/IEOF,B(11,2),TSTRT,MMNI,TF,W(10,2),NW(2),EPS(91) ,   CD5
     1ABSERR(31),XLNGTH ,TCPU,CONTROL(4)                                CD5
C**SET DEFAULTS                                                         INPL
      DATA DFYLU,IDF / 4.,1.E6,15.,0  /                                 INPL
      IF(IDF .EQ. 1)          GO TO 6                                   INPL
      MSBSTN=11                                                         INPL
      NR=0                                                              INPL
      B(1,1)=W(1,1)=1.E20                                               INPL
      NW(1)=1                                                           INPL
      DO 5 L = 1,3                                                      INPL
      YLU(1,L) = -DFYLU(L)                                              INPL
    5 YLU(2,L) =  DFYLU(L)                                              INPL
      IDF = 1                                                           INPL
    6 CONTINUE                                                          INPL
      RETURN                                                            INPL
C**ACCEPT DATA                                                          INPL
      ENTRY INPLTOT                                                     INPL
      IF (J .EQ. 21) GO TO 15                                           INPL
C**PLOT CARD                                                            INPL
      ENCODE(30,10,NAME)(BUFFER(L),L=1,30)                              INPL
 10   FORMAT(30R1)                                                      INPL
      CALL FIELDS(51B,6,BUFFER)                                         INPL
      NR=1                                                              INPL
      W(1,1)=FIELDA(4)                                                  INPL
      R(1)=FIELDA(5)                                                    INPL
      NREPLT = IF(6)                                                    INPL
      RETURN                                                            INPL
C**AXES CARD                                                            INPL
 15   CALL FIELDS(5252B,6,BUFFER)                                       INPL
      DO 20 L=1,6                                                       INPL
 20   YLU(L)=FIELDA(L)                                                  INPL
      RETURN                                                            INPL
C**DERIVED PLOTTING PARAMETERS                                          INPL
      ENTRY INPLFIN                                                     INPL
      IF(NR .EQ. 0)           GO TO 900                                 INPL
      IF(NREPLT .EQ. 3)       GO TO 800                                 INPL
      B(2,1) = TF                                                       INPL
      TTT(1)=B(1,1)=5.                                                  INPL
      TTT(2)=TF                                                         INPL
C**DATA ECHO                                                            INPL
  800 WRITE(6,25) NAME,CODE,PHONE,W(1,1),R(1),YLU                       INPL
 25   FORMAT(1H-,*PLOTTING PARAMETERS*,20X,3A10/E20.8,5X,*SECS BETWEEN P INPL
     1LOTTED VALUES*/E20.8,5X,*SECS PER FRAME*///5X,*AXES LIMITS*//     INPL
     2 T10,*PITCH*T20,2E20.8/T10,*MOSMP BNDG MMNT*,T20,2E20.8/          INPL
     2T10,*WAVE HEIGHT*,T20,2E20.8)                                     INPL
      RETURN                                                            INPL
 900  WRITE(6,910)                                                      INPL
 910  FORMAT(1H-,*NO PLOTTING*)                                         INPL
      RETURN                                                            INPL
      END                                                               INPL
```

Table 12 (Continued)

```
      SUBROUTINE INPRIN                                                 INPR     2
C                                                                       INPR     3
C**PRINTING TIMES                                                       INPR     4
C                                                                       INPR     5
      COMMON/FBLOK/FIELDA(6) ,IF(6)                                     CD2      2
      COMMON /HBLOK/ KAG(21),RINT(21),EI(20),SHPMS(20),DAMPC(20),       CD1      2
     1 DKI,NEVAL,SPEED,BUOY SPG(9),SCF(9),ADDMS(9),VRTMS(20),IBEG,IEND  CD1      3
      COMMON/INPT1/IEOF,B(11,2),TSTRT,HMNI,TF,W(10,2),NW(2),EPS(91) ,   CD5      2
     1ABSERR(91),XLNGTH ,TCPU,CONTROL(4)                                CD5      3
      COMMON /INPT2/J,ISFIELD,BUFFER(60),ICFIELD                        INPR     9
C**SET DEFAULTS (PART 1)                                                INPR    10
      NW(2)=0                                                           INPR    11
      B(1,2) = W(1,2) = 1.E20                                           INPR    12
      IBEG=IEND=-1                                                      INPR    13
      RETURN                                                            INPR    14
C**ACCEPT DATA                                                          INPR    15
      ENTRY INPRDT                                                      INPR    16
      IF (J .EQ. 16) GO TO 5                                            INPR    17
C**DEBUG                                                                INPR    18
      CALL FIELDS(240CB,6,BUFFER)                                       INPR    19
      IBEG=IF(1)                                                        INPR    20
      IEND=IF(2)                                                        INPR    21
      RETURN                                                            INPR    22
C**PRINT                                                                INPR    23
 5    CALL HSTDFN(BUFFER,10,NW(2),B(1,2),W(1,2),XMAX)                   INPR    24
      IN=NW(2)+1                                                        INPR    25
      B(IN,2)=XMAX                                                      INPR    26
      RETURN                                                            INPR    27
      ENTRY INPRFIN                                                     INPR    28
C**DATA ECHO                                                            INPR    29
 10   I1=NW(2)                                                          INPR    30
      WRITE(6,20) IBEG,IEND,(B(I,2),W(I,2),I=1,I1),B(I1+1,2)            INPR    31
 20   FORMAT(1H-,*DEBUG*/T10,*IBEG*,I10,5X,*IEND*I10/1H-,*PRINTING TIMES INPR    32
     1*,//T5,E15.5,3(*(*,E15.5,*)*,E15.5))                              INPR    33
      RETURN                                                            INPR    34
      END                                                               INPR    35
```

Table 12 (Continued)

```
      SUBROUTINE INTIN                                                  ININ       2
C                                                                       ININ       3
C**INITIAL CONDITIONS                                                   ININ       4
C                                                                       ININ       5
      COMMON /HBLOK/ KAG(21),RINT(21),EI(20),SHPMS(20),DAMPC(20),       CD1        2
     1 DXI,NEVAL,SPEED,BUOY SPG(9),SCF(9),ADDMS(9),VRTMS(20),IBEG,IEND  CD1        3
      COMMON/FBLOK/FIELDA(6) ,IF(6)                                     CD2        2
      COMMON/INPT1/IEOF,B(11,2),TSTRT,HMNI,TF,W(10,2),NW(2),EPS(91) ,   CD5        2
     1ABSERR(91),XLNGTH ,TCPU,CONTROL(4)                                CD5        3
      COMMON /INPT2/J,ISFIELD,BUFFER(60),ICFIELD                        ININ       9
      DIMENSION YS(91),SF(21),GDOT(21),BDM(20),YDOT(20),YH(9)           ININ      10
      COMMON //YS                                                       ININ      11
      EQUIVALENCE                                                       CD3        2
     1  (YS,YH),(YS(10),YDOT),(YS(30),GDOT),(YS(51),BDM),(YS(71),SF)    CD3        3
      DATA I2,I3,I4/2,3,4/                                              ININ      13
C**ALLOW I.C."S TO BE ON STATION, HALF-STATION, HYDROFORC CARDS         ININ      14
      CALL STIN (0)                                                     ININ      15
      CALL  HSTIN(0)                                                    ININ      16
      CALL HYDIN(0)                                                     ININ      17
      CALL INTRIN(0)                                                    ININ      18
      IST=IHS=IHY=0                                                     ININ      19
      RETURN                                                            ININ      20
C**TRANSGENERATED I.C."S DETECTED                                       ININ      21
      ENTRY INITDT                                                      ININ      22
      CALL FIELDS(3000B,6,BUFFER)                                       ININ      23
      KTYPE=IF(1)                                                       ININ      24
      FIELDA(1)=FIELDA(2)                                               ININ      25
      CALL FIELDS(12B,3,BUFFER)                                         ININ      26
      GO TO (10,20,30,40) KTYPE                                         ININ      27
 10   TSTRT=FIELDA(2)                                                   ININ      28
      CALL INTRIN(1)                                                    ININ      29
      RETURN                                                            ININ      30
 20   IF (IHS .EQ. 1) GO TO 25                                          ININ      31
      IHS=1                                                             ININ      32
      CALL HSTIN(1)                                                     ININ      33
 25   I=FIELDA(1)+.50001                                                ININ      34
      YDOT(I)=FIELDA(2)                                                 ININ      35
      BDM(I)=FIELDA(3)                                                  ININ      36
      RETURN                                                            ININ      37
 30   IF (IST .EQ. 1) GO TO 35                                          ININ      38
      IST=1                                                             ININ      39
      CALL STIN(1)                                                      ININ      40
 35   I=FIELDA(1)+.0001                                                 ININ      41
      GDOT(I)=FIELDA(2)                                                 ININ      42
      SF(I)=FIELDA(3)                                                   ININ      43
      RETURN                                                            ININ      44
 40   IF (IHY .EQ. 1) GO TO 45                                          ININ      45
      IHY=1                                                             ININ      46
      CALL HYDIN(1)                                                     ININ      47
 45   IF(1)=FIELDA(1)+.0001                                             ININ      48
      IF (IF(1) .LT. 3 .OR. IF(1) .GT. 19  .OR. (IF(1)/2)*2 .EQ. IF(1)) ININ      49
     1 CALL HERR                                                        ININ      50
      I=(IF(1)-1)/2                                                     ININ      51
      YH(I)=FIELDA(2)                                                   ININ      52
      RETURN                                                            ININ      53
C                                                                       ININ      54
C  PUNCH FINAL CONDITIONS                                               ININ      55
C                                                                       ININ      56
      ENTRY PUNCHIT                                                     ININ      57
      WRITE (7,50)  TSTRT                                               ININ      58
 50   FORMAT(*INITIAL*,T20,*1*,T31,E20.10)                              ININ      59
      DO 60 I=1,20                                                      ININ      60
      HS=I-.5                                                           ININ      61
 60   WRITE(7,70) I2,HS,YDOT(I),BDM(I)                                  ININ      62
 70   FORMAT(*INITIAL*,T11,I10,F10.1,2E20.10)                           ININ      63
      DO 80 I=1,21                                                      ININ      64
      IF (I .GT. 9) GO TO 75                                            ININ      65
      II=2*I+1                                                          ININ      66
      WRITE(7,90) I4,II,YH(I)                                           ININ      67
 75   II=I-1                                                            ININ      68
 80   WRITE(7,90) I3,II,GDOT(I),SF(I)                                   ININ      69
 90   FORMAT(*INITIAL*,T11,2I10,2E20.10)                                ININ      70
      END                                                               ININ      71
```

Table 12 (Continued)

```
      SUBROUTINE INTRIN(ITS)
C
C**INTEGRATION DATA
C
      COMMON/FBLOK/FIELDA(6) ,IF(6)
      COMMON/INPT1/IEOF,B(11,2),TSTRT,HMNI,TF,W(10,2),NW(2),EPS(91) ,
     1ABSERR(91),XLNGTH ,TCPU,CONTROL(4)
      COMMON /INPT2/J,ISFIELD,BUFFER(60),ICFIELD
      DATA KX/0/
      IF (ITS .EQ. 0) GO TO 1
      ITC=1
      RETURN
C**SET DEFAULT VALUES
 1    ITC=0
      IF (KX .EQ. 1) RETURN
      KX=1
      HMNI=TSTRT=TF=TCPU=0.
      RETURN
C**ACCEPT DATA
      ENTRY INTRDT
      ISFIELD=ISFIELD+1
      IF (ISFIELD .NE. 1) GO TO 20
C**FIRST CARD
 10   CALL FIELDS(5240B,6,BUFFER)
      HMNI=FIELDA(1)
      TF=FIELDA(3)
      TCPU=FIELDA(4)
      IF (ITC .EQ. 1) GO TO 29
      TSTRT=FIELDA(2)
      GO TO 29
C**SECOND CARD
 20   CALL FIELDS(5250B,6,BUFFER)
      IF (ISFIELD .EQ. 3)     GO TO 21
      DO 35 I = 1,20
      IF (I .GT. 9) GO TO 33
      EPS(I)=FIELDA(1)
 33   EPS(I+9)=FIELDA(2)
      EPS(I+29)=FIELDA(3)
      EPS(I+50)=FIELDA(4)
 35   EPS(I+70)=FIELDA(5)
      EPS(50)=FIELDA(3)
      EPS(91)=FIELDA(5)
 29   ICFIELD=1
      RETURN
C**THIRD CARD
 21   DO 25 I = 1,20
      IF (I .GT. 9) GO TO 23
      ABSERR(I)=FIELDA(1)
 23   ABSERR(I+9)=FIELDA(2)
      ABSERR(I+29)=FIELDA(3)
      ABSERR(I+50)=FIFLDA(4)
 25   ABSERR(I+70)=FIELDA(5)
      ABSERR(50)=FIELDA(3)
      ABSERR(91)=FIELDA(5)
      ICFIELD=0
      RETURN
C**DATA ECHO
      ENTRY INTRFIN
      WRITE (6,200) HMNI,TSTRT,TF,TCPU,
     1           EPS(1),    EPS(10),    EPS(30),    EPS(51),    EPS(71),
     2           ABSERR(1),ABSERR(10),ABSERR(30),ABSERR(51),ABSERR(71)
 200  FORMAT(1H-,*INTEGRATION DATA*//T10,*MAXIMUM INTERNAL STEP SIZE (HM
     1NI)*,T50,E20.10/T10,*SIMULATION START (TSTRT)*,T50,E20.10/T10,*SIM
     3ULATION END (TF)*,T50,E20.10/T10,*CPU SECS*,T50,E20.10///T5,*MAXIM
     1UM ERRORS*/T25,*VERTICAL*,
     4 T40,*VERTICAL*,T60,*ANGULAR*,T80,*BENDING*,T100,*SHEAR*/T20,*DISP
     5LACEMENT*,T40,*VELOCITY*,T60,*VELOCITY*,T80,*MOMENT*,T100,*FORCE*/
     6T5,*RELATIVE*,T15,5E20.10/T5,*ABSOLUTE*,T15,5E20.10)
      RETURN
      END
```

Table 12 (Continued)

```
      SUBROUTINE MSCON                                                  INMS       2
C                                                                       INMS       3
C  MISCELLANEOUS DATA--                                                 INMS       4
C       SPEED                                                           INMS       5
C       SECTION LENGTH                                                  INMS       6
C                                                                       INMS       7
      COMMON/FBLOK/FIELDA(6) ,IF(6)                                     CD2        2
      COMMON /HBLOK/ KAG(21),RINT(21),EI(20),SHPMS(20),DAMPC(20),       CD1        2
     1 DXI,NEVAL,SPEED,BUOY SPG(9),SGF(9),ADDMS(9),VRTMS(20),IBEG,IEND  CD1        3
      COMMON/INPT1/IEOF,B(11,2),TSTRT,HMNI,TF,W(10,2),NH(2),EPS(91) ,   CD5        2
     1ABSERR(91),XLNGTH ,TCPU,CONTROL(4)                                CD5        3
      COMMON /INPT2/J,ISFIELD,BUFFER(60),ICFIELD                        INMS      11
C**SET DEFAULT VZLUES                                                   INMS      12
      SPEED=0.                                                          INMS      13
      DXI=1.                                                            INMS      14
      RETURN                                                            INMS      15
C**ACCEPT DATA                                                          INMS      16
      ENTRY MSCDT                                                       INMS      17
      CALL FIELDS(4000B,6,BUFFER)                                       INMS      18
      IF (J .EQ. 19) GO TO 10                                           INMS      19
C**SPEED                                                                INMS      20
      SPEED=FIELDA(1)                                                   INMS      21
      RETURN                                                            INMS      22
C**SECTION LENGTH                                                       INMS      23
 10   DXI=FIELDA(1)                                                     INMS      24
      XLNGTH = 16.*DXI                                                  INMS      25
      RETURN                                                            INMS      26
C**DATA ECHO                                                            INMS      27
      ENTRY MSFIN                                                       INMS      28
      WRITE(6,20) SPEED,DXI                                             INMS      29
 20   FORMAT(1H-,*MISCELLANEOUS DATA*//T10,*SPEED = *,T20,E15.7/T10,    INMS      30
     1                                                   *SECTION LENG INMS      31
     1TH = *,T30,E15.7)                                                 INMS      32
      RETURN                                                            INMS      33
      END                                                               INMS      34
```

Table 12 (Continued)

```
      SUBROUTINE STIN(ITS)
C
C**STATION CARDS
C
      REAL KAG
      COMMON /HBLOK/ KAG(21),RINT(21),EI(20),SHPMS(20),DAMPC(20),
     1 DXI,NEVAL,SPEED,BUOY SPG(9),SCF(9),ADDMS(9),VRTMS(20),IBEG,IEND
      COMMON /INPT2/J,ISFIELD,BUFFER(60),ICFIELD
      DIMENSION YS(91),SF(21),GDOT(21),BDM(20),YDOT(20),YH(9)
      COMMON //YS
      COMMON/FBLOK/FIELDA(6) ,IF(6)
      EQUIVALENCE
     1  (YS,YH),(YS(10),YDOT),(YS(30),GDOT),(YS(51),BDM),(YS(71),SF)
      DATA KX /0/
      IF (ITS .EQ. 0) GO TO 1
      ITC=1
      RETURN
C**SET DEFAULT VALUES
 1    ITC=0
      IF (KX .EQ. 1) RETURN
      KX=1
      DO 5 I=1,21
 5    KAG(I)=RINT(I)=SF(I)=GDOT(I)=.0
      RETURN
C**ACCEPT DATA
      ENTRY STNDT
      CALL FIELDS(32508,6,BUFFER)
      I=IF(1)+1
      KAG(I)=FIELDA(2)
      RINT(I)=FIELDA(3)
      IF (ITC .EQ. 1) RETURN
      GDOT(I)=FIELDA(4)
      SF(I)=FIELDA(5)
      RETURN
C**DATA ECHO
      ENTRY STFIN
      WRITE(6,10)
 10   FORMAT(1H1,*STATION DATA*///T10,*KAG = SHEAR STIFFNESS*/
     1 T10,*RINT = MOMENT OF INERTIA*/
     1 T10,*GDOT = INITIAL ANGULAR VELOCITY*/T10,*SF = INITIAL SHEAR FOR
     2CE*
     2////1X,*STATION*,T19,*KAG*,T34,*RINT*,T49,*GDOT*,T64,
     3 *SF*)
      DO 20 L=1,21
      I=L-1
 20   WRITE(6,30) I,KAG(L),RINT(L),GDOT(L),SF(L)
 30   FORMAT(1X,I5,10X,4E15.7)
      RETURN
      END
```

Table 12 (Continued)

```
      FUNCTION IPWR2(J)
C  VALUE RETURNED IS SMALLEST POWER OF 2 GREATER THAN OR EQUAL TO *J*
      JJ=J+1
      IPWR2=1
 1    CONTINUE
      IF(IPWR2 .GE. JJ) RETURN
      IPWR2=IPWR2*2
      GO TO 1
      END
```

```
      SUBROUTINE ISEARCH(X,CLSMRK,NUCMS,IPP,IL,ISTATE)
C  ARRAY *CLSMRK* OF SIZE *NUCMS* IS SEARCHED FOR *X*.  IF FOUND,
C  *ISTATE* IS SET TO 1, AND *IL* IS SUCH THAT  *CLSMRK(IL)* IS EQUAL
C  TO *X*.  IF NOT FOUND, *ISTATE* IS ZEROED AND *IL* IS SUCH THAT
C  *CLSMRK(IL)* .LT. *X*, PLUS *CLSMRK(IL+1)*  .GT. *X* FOR  *IL* .NE.
C  *NUCMS*.
      INTEGER CLSMRK,X
      DIMENSION CLSMRK(50)
      KKK=0
      IF (NUCMS.EQ. 0) GO TO 700
      J=IPP/2
      K=IPP/4
      IF(J .GT. NUCMS) GO TO 20
 9    IF (X .GE. CLSMRK(J)) GO TO 10
 20   IF(K.EQ. 0) GO TO 41
      J=J-K
      K=K/2
      IF(J.GT.NUCMS) GO TO 20
      GO TO 9
 10   IF (K .EQ. 0) GO TO 31
      J=J+K
      K=K/2
      IF ( J .GT. NUCMS) GO TO 20
      GO TO 9
 31   KKK=J
      GO TO 51
 41   KKK=J-1
 51   IF(KKK .EQ. 0) GO TO 700
      IF(X .NE. CLSMRK(KKK)) GO TO 700
      ISTATE=1
      IL=KKK
      RETURN
 700  ISTATE=0
      IL=KKK+1
      RETURN
      END
```

Table 12 (Continued)

```
      REAL FUNCTION NOLEX(NTYPE,C,X)
C$    DEBUG
C$    STORES(NOLEX)
      DIMENSION C(1)
      COMMON/NBLOCK/DX
      GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18) NTYPE
 1    NOLEX =C(1) $ RETURN
 2    NOLEX=C(1)*X     $ RETURN
 3    NOLEX=C(1)*DX $ RETURN
 4    NOLEX=C(1)*X+C(2) $ RETURN
 5    NOLEX=C(1)*X**2 $ RETURN
 6    NOLEX=C(1)*DX**2 $ RETURN
 7    NOLEX=X*(C(1)*X+C(2)) $ RETURN
 8    NOLEX=X*(C(1)*X+C(2))+C(3)                 $ RETURN
 9    NOLEX=C(1)*X**3 $ RETURN
 10   NOLEX=C(1)*DX**3 $ RETURN
 11   NOLEX=X**2*(C(1)*X+C(2)) $ RETURN
 12   NOLEX=X*(X*(C(1)*X+C(2))+C(3)  )
 13   NOLEX=X*(X*(C(1)*X+C(2))+C(3))+C(4) $ RETURN
 14   NOLEX=C(1)*X**2+C(2) $ RETURN
 15   NOLEX=X*(C(1)*X**2+C(2)) $ RETURN
 16   NOLEX=X*(C(1)*X**2+C(2))+C(3) $ RETURN
 17   NOLEX=C(1)*X**3+C(2) $ RETURN
 18   NOLEX=X**2*(C(1)*X+C(2))+C(3) $ RETURN
      END
```

Table 12 (Continued)

```
      SUBROUTINE SCAN (ISTRING,ISTART,LENGTH,KEWDARY,NUKEWD,KEYCODE,ILAS
     1T)
C-- ISTRING AND KEWDARY MUST BE RIGHT JUSTIFIED WITH ZERO FILL
      DIMENSION ISTRING(1),IALA(10)
      DIMENSION KEWDARY(1)
      IF(ISTART .GT. LENGTH .OR. NUKEWD .LE. 0) GO TO 150
      KK=0
      J=0
      DO 100 I=ISTART,LENGTH
      IF (ISTRING(I) .NE. 1RG .AND. ISTRING(I) .NE. 1RL) GO TO 100
      J=J+1
      KK=ISTRING(I)+KK*64
  100 CONTINUE
      DO 65 L=1,NUKEWD
      IF(KK .EQ. KEWDARY(L)) GO TO 200
   65 CONTINUE
  150 ILAST=LENGTH    +ISTART
      KEYCODE=0
      RETURN
  200 ILAST=I
      KEYCODE=L
      RETURN
      END
```

```
      FUNCTION POLY(A,X)
      DIMENSION A(1)
      N = IFIX(A(1)) + 1
      J = N + 2
      POLY = 0.
      DO 400 I=1,N
      J = J - 1
  400 POLY = POLY * X + A(J)
      RETURN
      END
```

Table 12 (Continued)

```
INPUT CARDS
COMMENT    ESSEX    DISCRETE WAVE TRAIN
SEA-DATA   1.08470672 .598444   .804804    4.11238748 -21.97272 -29.54952
CONTROL    PUNCH =      NO       NL HYDRO =    NO      HYD DAMP=    YES
PRINT      0.          (1.)      50.        $
INTGRTN    .015625    0.         50.        1500.                             +INT1
+INT1      .1         .01        .01        .2         .05                    +INT2
+INT2      1.         1.         .2         2000.      300.
PLOT       SWYBRANIEC1844        71940      8.1        50.        2           0.
AXES       -3.        3.         -1.+6      1.+6       -12.       12.         0.
DAMPC      .750       .023
SPEED      65.
SCTN LNGTH 41.
STATION    0          1339870.   26.86
STATION    1          1636730.   254.08
STATION    2          2484850.   419.05
STATION    3          3531440.   615.63
STATION    4          4770220.   830.89
STATION    5          6056130.   1015.57
STATION    6          7142850.   1157.16
STATION    7          7765150.   1267.88
STATION    8          7831900.   1337.42
STATION    9          7509160.   1341.24
STATION    10         7180390.   1290.96
STATION    11         6966860.   1234.65
STATION    12         7020550.   1169.07
STATION    13         7529840.   1061.59
STATION    14         8151090.   910.29
STATION    15         7321430.   736.49
STATION    16         4000000.   546.97
STATION    17         2455090.   354.50
STATION    18         1855200.   203.79
STATION    19         1431060.   130.25
STATION    20         1269350.   84.31
HALF-STTN .5          .45467     5.05075+9  .018137
HALF-STTN 1.5         .74401     7.76725+9  .030892
HALF-STTN 2.5         .84735     12.31675+9 .085016
HALF-STTN 3.5         1.13281    18.30495+9 .096434
HALF-STTN 4.5         1.70245    27.046 +9  .17205
HALF-STTN 5.5         2.10287    36.95395+9 .18807
HALF-STTN 6.5         2.20362    44.24075+9 .22888
HALF-STTN 7.5         2.14550    49.67695+9 .22655
HALF-STTN 8.5         2.09512    55.16235+9 .24224
HALF-STTN 9.5         2.10158    58.75385+9 .2425
HALF-STTN 10.5        2.04374    59.00065+9 .2317
HALF-STTN 11.5        1.99308    50.67175+9 .22967
HALF-STTN 12.5        1.94270    37.47675+9 .19103
HALF-STTN 13.5        1.85357    33.20715+9 .16746
HALF-STTN 14.5        1.76574    34.4828 +9 .13112
HALF-STTN 15.5        1.65465    28.64295+9 .12667
HALF-STTN 16.5        1.35756    19.27405+9 .074156
HALF-STTN 17.5        .8706      13.65365+9 .054678
```

Table 12 (Continued)

```
HALF-STTN 18.5       .54122      9.82855 +9 .025649
HALF-STTN 19.5       .36426      7.4482  +9 .018570
HYDRO      3         1.27805     2.605691  .9077      .9
HYDRO      5         2.59878     3.184146  2.006      1.5
HYDRO      7         3.518293    3.295122  2.6419     1.5
HYDRO      9         3.960976    3.301220  2.8519     1.5
HYDRO      11        3.74878     3.273171  2.7917     1.5
HYDRO      13        2.832927    3.032927  2.3768     1.7
HYDRO      15        1.512195    2.321951  1.592      1.5
HYDRO      17        .496341     1.235366  .8216      .53
HYDRO      19        .1          .340244   .3184      .05
NL1B          3        0.                                                 GG1
+G1          G         5       .05179                                     GG2
+G2          L         1        0.
NL1BUOYANCY       15       0.                                             BB1
+B1        YGYC        5           .02512                                 BB2
+B2           L          1      0.
NLBUOYANCY  17           5      .01345
NL1BUOYANCY      19       -10                                             AA1
+A1        YLYC           6     .03024                                    AA2
+A2        YGYC           1       0
NL2ADDMASS    3         0.        0.                                      FF1
+F1          LG         2      -.03732                                    FF2
+F2          GG       2        -.06829                                    FF3
+F3        L L        4          -.01474  .5602                           FF4
+F4         G L          4     -.09154     .5602
NL1ADDMASS     15        0.                                               EE1
+E1          G         7      -.7805E-03-26.13-3                          EE2
+E2          L        2        -26.13E-3
NL2ADDMASS  17        0.          0.                                      DD1
+D1          LEG         7     1.568-03   -13.17E-3                       DD2
+D2          G G           2    -13.17E-3                                 DD3
+D3           LL        7       2.72E-3    -13.17E-3                      DD4
+D4                GL       2   -13.17E-3
NL2ADDMASS          19     -10.       -0                                  CC1
+C1           LG         10    -.0001476                                  CC2
+C2           GG        1         -0.                                     CC3
+C3          L L         10    -.2658E-3                                  CC4
+C4           GAL         1       0
```

Table 12 (Continued)

STATION DATA

KAG = SHEAR STIFFNESS
RINT = MOMENT OF INERTIA
GDOT = INITIAL ANGULAR VELOCITY
SF = INITIAL SHEAR FORCE

STATION	KAG	RINT	GDOT	SF
0	.1339870E+07	.2686000E+02	0.	0.
1	.1636730E+07	.2540800E+03	0.	0.
2	.2484850E+07	.4190500E+03	0.	0.
3	.3531440E+07	.6156300E+03	0.	0.
4	.4770220E+07	.8308900E+03	0.	0.
5	.6056130E+07	.1015570E+04	0.	0.
6	.7142860E+07	.1157160E+04	0.	0.
7	.7765150E+07	.1267880E+04	0.	0.
8	.7831900E+07	.1337420E+04	0.	0.
9	.7509160E+07	.1341240E+04	0.	0.
10	.7180390E+07	.1290960E+04	0.	0.
11	.6966860E+07	.1234650E+04	0.	0.
12	.7020550E+07	.1169070E+04	0.	0.
13	.7529840E+07	.1061590E+04	0.	0.
14	.8151090E+07	.9102900E+03	0.	0.
15	.7321430E+07	.7364900E+03	0.	0.
16	.4000000E+07	.5469700E+03	0.	0.
17	.2455090E+07	.3549000E+03	0.	0.
18	.1855200E+07	.2037900E+03	0.	0.
19	.1431060E+07	.1302500E+03	0.	0.
20	.1269350E+07	.8431000E+02	0.	0.

STRUCTURAL DAMPING PARAMETERS
NATURAL FREQUENCY .7500000000E+00 HZ
PERCENT DAMPING .2300000000E-01

HALF STATION DATA

SHPMS = MASS PER UNIT LENGTH
EI = BENDING RIGIDITY
DAMPC = STRUCTURAL DAMPING COEFICIENT
YDOT = INITIAL VERTICAL VELOCITY
BOM = INITIAL BENDING MOMENT

Table 12 (Continued)

HALF-STATION	SHPMS	EI	DAMPC	YDOT	BOM
.5	.4546700E+00	.5050750E+10	.1568612E-01	0.	0.
1.5	.7440100E+00	.7767250E+10	.6976107E-01	0.	0.
2.5	.8473500E+00	.1231675E+11	.7332630E-01	0.	0.
3.5	.1132810E+01	.1830495E+11	.8317467E-01	0.	0.
4.5	.1702450E+01	.2704600E+11	.1483924E+00	0.	0.
5.5	.2102870E+01	.3695395E+11	.1622069E+00	0.	0.
6.5	.2203620E+01	.4424075E+11	.1974060E+00	0.	0.
7.5	.2145500E+01	.4967695E+11	.1954009E+00	0.	0.
8.5	.2095120E+01	.5516235E+11	.2089353E+00	0.	0.
9.5	.2101580E+01	.5875365E+11	.2091582E+00	0.	0.
10.5	.2043740E+01	.5900065E+11	.1998419E+00	0.	0.
11.5	.1993080E+01	.5067175E+11	.1960942E+00	0.	0.
12.5	.1942700E+01	.3747675E+11	.1647591E+00	0.	0.
13.5	.1853570E+01	.3320715E+11	.1616841E+00	0.	0.
14.5	.1765740E+01	.3448280E+11	.1130888E+00	0.	0.
15.5	.1654650E+01	.2864295E+11	.1092562E+00	0.	0.
16.5	.1357560E+01	.1927405E+11	.6395958E-01	0.	0.
17.5	.8706000E+00	.1365365E+11	.4715946E-01	0.	0.
18.5	.5412200E+00	.9828550E+10	.2212209E-01	0.	0.
19.5	.3642600E+00	.7448200E+10	.1601697E-01	0.	0.

INTEGRATION DATA

MAXIMUM INTERNAL STEP SIZE (HMNI)	.1562500000E-01
SIMULATION START (TSTRT)	0.
SIMULATION END (TF)	.5000000000E+02
CPU SECS	.1500000000E+04

MAXIMUM ERRORS

	VERTICAL DISPLACEMENT	VERTICAL VELOCITY	ANGULAR VELOCITY	BENDING MOMENT	SHEAR FORCE
RELATIVE	.1000000000E+00	.1000000000E-01	.1000000000E-01	.2000000000E+00	.5000000000E-01
ABSOLUTE	.1000000000E+01	.1000000000E+01	.2000000000E+00	.2000000000E+04	.3000000000E+03

MISCELLANEOUS DATA

SPEED = .6500000E+02
SECTION LENGTH = .4100000E+02

HYDRO-FORCE DATA

ADDMS = LINEAR ADDED MASS
BUOY = LINEAR BYOYANCY
SCF = SMITH CORRECTION
HDMP = HYDRO-DAMPING
YH = INITIAL VERTICAL DISPLACEMENT

Table 12 (Continued)

STATION	ADDMS	BUOY	SCF	HDMP	YH
3	.1278050E+01	.2605691E+01	.9077000E+00	.9000000E+00	0.
5	.2598780E+01	.3184146E+01	.2006000E+01	.1500000E+01	0.
7	.3518293E+01	.3295122E+01	.2641900E+01	.1500000E+01	0.
9	.3960976E+01	.3301220E+01	.2651900E+01	.1500000E+01	0.
11	.3748780E+01	.3273171E+01	.2791700E+01	.1500000E+01	0.
13	.2832927E+01	.3032927E+01	.2376800E+01	.1700000E+01	0.
15	.1512195E+01	.2321951E+01	.1592000E+01	.1500000E+01	0.
17	.4963410E+00	.1235366E+01	.8216000E+00	.5300000E+00	0.
19	.1000000E+00	.3402440E+00	.3184000E+00	.5000000E-01	0.

```
CONTROL OPTIONS
        PUNCH FINAL CONDITIONS          NO
        NON-LINEAR HYDRO FORCE          NO
        HYDRO DAMPING                   YES

PLOTTING PARAMETERS                      SHYBRANIEC1844      71943
      .10000000E+00    SECS BETWEEN PLOTTED VALUES
      .50000000E+02    SECS PER FRAME

    AXES LIMITS

        PITCH          -.30000000E+01       .30000000E+01
        HDSHP BNDG     -.10000000E+07       .10000000E+07
        WAVE HEIGH     -.12000000E+02       .12000000E+02

PLOTTING PARAMETERS

DEBUG
        IBEG        -1        IEND        -1

PRINTING TIMES

      0.          (     .10000E+01)     .50000E+02

SEAGEN IS USING DISCRETE WAVE TRAIN
```

Table 12 (Continued)

```
INPUT DATA

     DRAFT = SHIPS DRAFT =     .100000E+02 FT
     DENS = MASS DENSITY OF WATER =    .200000E+01 SLUGS/CU.FT.
     BTANG = ALPHA, BUTTOCK ANGLE =  0.           DEGREES
     DRANG = BETA, DEADRISE ANGLE =    .300000E+01 DEGREES
     DX = STATION LENGTH =    .410000E+02 FT
     BSLP = SLOPE RELATED TO WIDTH ARRAY =    .100000E+00

PRE-COMPUTED DATA

     SINAL = SINE OF ALPHA =  0.
     COSAL = COSINE OF ALPHA =    .100000E+01
     TANAL = TANGENT OF ALPHA =  0.
     COSBET = COSINE OF BETA =    .998630E+00
     TANBET = TANGENT OF BETA =    .524078E-01
     DX2 = 2. * STATION LENGTH =    .820000E+02 FT
     DX10 = 10. * STATION LENGTH =    .410000E+03 FT
     DNSBY2 = MASS DENSITY OF WATER / 2. =    .100000E+01 SLUGS/CU.FT.

          HEIGHT OF DEADRISE              WIDTH OF SLAM REGION
             DRHGT(10)                        WIDTH(10)
             .241076E+01 FT                 .460000E+02 FT
             .241076E+01 FT                 .460000E+02 FT
             .241076E+01 FT                 .460000E+02 FT
             .241076E+01 FT                 .460000E+02 FT
             .241076E+01 FT                 .460000E+02 FT
             .241076E+01 FT                 .460000E+02 FT
             .241076E+01 FT                 .460000E+02 FT
             .241076E+01 FT                 .460000E+02 FT
             .241076E+01 FT                 .460000E+02 FT
             .241076E+01 FT                 .460000E+02 FT

           COEF(8,1)             COEF(8,2)             COEF(8,3)
          .200000E+01           .500000E+01           .500000E+01
          .320000E+00           .218209E+01           .474874E+01
          .854664E+01          -.541549E+02          -.770645E+02
          .150461E+03           .668885E+03           .517540E+03
         0.                    -.439937E+04          -.174879E+04
         0.                     .146328E+05           .294746E+04
         0.                    -.193390E+05          -.197505E+04
         0.                    0.                    0.
```

Table 12 (Continued)

```
TIME    25.000000              20235 FCN EVALS
WV HGT     .209510E+01  .580456E+01  .674629E+01  .477054E+01  .231953E+00 -.517776E+01 -.911319E+01 -.964912E+01 -.623443E+01
WV VEL     .366895E+01  .239626E+01 -.501813E+00 -.381372E+01 -.592360E+01 -.561665E+01 -.273463E+01  .165536E+01  .568915E+01
REL VEL   -.850287E+00 -.109518E+01  .314247E+00  .217737E+01  .288405E+01  .123813E+01 -.290789E+01 -.848482E+01 -.136738E+02
REL DISP  -.807934E-01 -.397971E+01 -.513114E+01 -.338568E+01  .907192E+00  .606785E+01  .976407E+01  .100770E+02  .645522E+01
WAVE ACC   .240909E+00 -.245826E+01 -.373185E+01 -.329713E+01 -.114352E+01  .185680E+01  .432788E+01  .504491E+01  .354574E+01
RL WV VL   .224032E+01  .906477E+00  .798079E-01 -.215867E+00 -.862781E+00 -.227222E+01 -.449690E+01 -.717339E+01 -.108028E+02
ADD MASS   .127805E+01  .259878E+01  .351829E+01  .396098E+01  .374878E+01  .283293E+01  .151220E+01  .496341E+00  .100000E+00
N-L A MS   .210523E+00  .126720E+02  .169077E+02  .111769E+02 -.296939E+01 -.184033E+02 -.226717E+02 -.124488E+02 -.219635E+01
SPC DRV   -.439870E+00  .128465E+02  .201986E+02  .138194E+02 -.657728E+01 -.263198E+02 -.273590E+02 -.122997E+02 -.641690E+00
SCF       -.114215E+00  .605752E+01  .626799E+01 -.423831E-02 -.111234E+02 -.207012E+02 -.202548E+02 -.945901E+01 -.195416E+00
HYD DMP   -.213050E+01  .469781E+01  .614978E+01  .319563E+00 -.982922E+01 -.168384E+02 -.135094E+02 -.565712E+01  .344726E+00
SLM FRC    .140000E+02  .170000E+02  .106390E+03  .327671E+01  .542238E+02  .858718E+01  .976407E+01  .100770E+02  .735331E+01
 .766628E+01  .700000E+01  .800000E+01
SMOOTH    0.           0.           0.           .197822E+00  .589673E+00  .154132E+00 0.                        .177019E-02  .969519E+00
 .339672E+01  .134619E+01 0.
HYD FRC   0.          -.213050E+01 -.213050E+01 -.213050E+01  .469781E+01  .469781E+01  .614978E+01  .614978E+01  .319563E+00
 .319563E+00 -.982922E+01 -.982922E+01 -.166406E+02 -.162488E+02 -.133553E+02 -.135094E+02 -.565535E+01 -.468760E+01
 .374144E+01  .169091E+01
VRT MS     .212540E+01  .430123E+01  .572191E+01  .605610E+01  .579252E+01  .477563E+01  .327794E+01  .185390E+01  .641220E+00


TIME       25.00000 SECS         183.822 SECS CPU TIME
TIME STEP   .08625 SECS            20239 DERIVATIVE EVALUATIONS

VERTICAL POSITION
  2.0143E+00  1.8248E+00  1.6152E+00  1.3849E+00  1.1391E+00  8.9009E-01  6.5087E-01  4.2792E-01  2.2079E-01

VERTICAL VELOCITY
  4.7206E+00  3.9647E+00  3.1971E+00  2.4403E+00  1.6744E+00  9.2778E-01  1.7934E-01 -5.5446E-01 -1.2803E+00 -1.9919E+00
 -2.6947E+00 -3.3844E+00 -4.0509E+00 -4.7062E+00 -5.3381E+00 -5.9469E+00 -6.5480E+00 -7.1109E+00 -7.6915E+00 -8.2778E+00

PITCH            -1.566E-01

ANGULAR VELOCITY
 -1.8396E-02 -1.8396E-02 -1.8403E-02 -1.8321E-02 -1.8252E-02 -1.8089E-02 -1.7964E-02 -1.7752E-02 -1.7581E-02 -1.7293E-02
 -1.7110E-02 -1.6800E-02 -1.6455E-02 -1.6109E-02 -1.5460E-02 -1.5441E-02 -1.4746E-02 -1.4684E-02 -1.4711E-02 -1.4242E-02
 -1.4242E-02

BENDING MOMENT
  0.         -2.1804E+03 -1.3059E+04 -3.2264E+04 -5.8964E+04 -8.2101E+04 -9.7899E+04 -1.0300E+05 -9.3476E+04 -7.5509E+04
 -4.4135E+04 -1.4764E+04  1.5018E+04  3.4695E+04  4.1025E+04  4.2824E+04  3.0499E+04  1.8687E+04  5.3852E+03  0.

SHEAR FORCE
  0.         -5.1617E+01 -2.6395E+02 -4.6475E+02 -6.4469E+02 -5.5276E+02 -3.7898E+02 -1.0479E+02  2.2813E+02  4.6577E+02
  7.5320E+02  7.2708E+02  7.3892E+02  4.5566E+02  2.1740E+02 -2.9350E+01 -2.4484E+02 -2.9267E+02 -3.4257E+02 -1.1681E+02
  0.
```

TABLE 13 - PRINCIPAL FORTRAN VARIABLES FOR SUBROUTINE HYD FRC

Call Letters:	*Definition*	*Symbol*
ADDMS(I)	Added mass at still waterline	m_0
BUOYSPG(I)	Buoyancy spring	$K_b = \rho g b_1 \Delta X$
CEL	Celerity of wave propagation	c
DXI	Length of one station	ΔX
EI(I)	Bending rigidity	EI
HF(I)	Total hydrodynamic force	P
HFTEMP(I)	Intermediate sums of hydrodynamic force terms	
HYDDAMP(I)	Real part of hydrodynamic damping coefficient	$C(\omega)$
NAPPLY(J)	Array giving station number at which Jth hydrodynamic force is applied	
NCALC	Number of stations at which hydrodynamic force is calculated	
NLINV(J)	Station number at which Jth nonlinear hydrodynamic force is computed	
SCF(I)	*Smith correction factor*	$\rho A_0 \Delta X/m_0$
SHPMS(I)	Ship mass	m_s, m
SPEED	Forward ship speed	U
T	Time coordinate	t
VCRIT(J)	Relative velocity V_r at which nonlinear force changes expression for Jth nonlinear group	
VMTEMP(I)	Intermediate sums of mass terms (ship- plus added-mass)	
VR(I)	Vertical velocity of ship relative to wave	V_r
VRTMS	Total mass (ship- plus added-mass)	$m_s + m_V$
YCRIT(J)	Relative displacement Y_r at which nonlinear force changes expression for Jth nonlinear group	
YH	Vertical displacement of ship	Y
YHDOT	Vertical velocity of ship	$\dot{Y}$
YR	Vertical displacement of ship relative to wave	Y_r
YRDOT	Partial time derivative of Y_r	$\partial Y_r/\partial t$

TABLE 13 (Continued)

Call Letters:	*Definition*	*Symbol*
YW	Height of wave surface	Y_w
YWDOT	Vertical velocity of wave surface	$\dot{Y}_w$
YWDD	Vertical acceleration of wave surface	$\ddot{Y}_w$
ZBY4DXI		$U/4\Delta X$

TABLE 14 - INPUT CARD TYPES

Type:	*Logical Card Count Should Equal*	*Action if One or More Omitted*
HYDRØ	Number of calculated hydroforce stations (9)	At omitted stations ADDMS=BUOY=SCF=HDMP =YH=0
NL_A	Number of nonlinear added mass stations ($\leq$9)	At omitted stations added mass is linear
NL_B	Number of nonlinear buoyancy stations ($\leq$9)	At omitted stations buoyancy is linear
CØMMENT	0 or more	None
SCTN	0 or 1	DXI=1
SPEED	0 or 1	SPEED=0
CØNTRØL	0 or 1	$CNTRL_2$=$CNTRL_3$=$CNTRL_4$=2HNØ
DEBUG	0 or 1	IBEG=IEND=-1
INTGRTN	0 or 1	HMNI=TSTRT=TF=TCPU=0
AXES	0 or 1	$P_\ell=-5$, $P_u=5$, $W_\ell=-10$, $W_u=10$, $B_\ell=-5.E6$ $B_u=5.E6$
PLØT	0 or 1	No plots
PRINT	0 or 1	No printed output
SEA-GEN (Case 1)	0 or 1	FREQ=WVHGH=0
DAMPC	0 or 1	Structural damping coefficient on HALF-STTN cards are used
HALF-STTN	Number of half-stations (20)	At omitted half-stations SHPMS=EI=DAMPC =YDØT=BDM=0

TABLE 14 (Continued)

Type:	*Logical Card Count Should Equal*	*Action if One or More Omitted*
INITIAL	0 or more	Initial conditions on corresponding INTGRTN, HALF-STTN, STATIØN, or HYDRØ card are used if columns 11 through 20 of omitted card contain 1, 2, 3, or 4, respectively
STATIØN	Number of stations (21)	At omitted stations KAG=RINT=SF=GDØT=0

Input cards comprise one or more physical records. In a case input deck, physical records may appear in any order. Columns 1 through 10 of the initial record of a card contain the card-type name. Input cards consisting of more than one physical record require a unique continuation field in columns 72 through 80 of all its nonterminal records. On noninitial records, column 1 contains a +, and columns 2 through 10 match the continuation field of the logically preceding record.

Numeric input requirements are less strict than I and E, i.e., integer and exponent, formats in that

1. Integer fields need not be right justified
2. Nonnumeric characters are ignored, except for the decimal point, e.g., blank and comma
3. Real fields lacking a decimal point are treated as fractionless real numbers
4. Integer fields containing a decimal point are truncated.

In using the program ROSAS, the following input cards are needed and are explained in detail.

ΔΔΔ1. Slamming Data

Seven data cards are needed for SLAM. They must come first in the data deck and be in the following order.

Card 1:

1–10	10–20	20–30	30–40	40–80
DRAFT	DENS	BTANG	DRANG	Ignored

Field:	*Format*	*Contents*
DRAFT	F10.0	Draft of ship in feet
DENS	F10.0	Mass density of water in slugs per cubic foot
BTANG	F10.0	Buttock angle in degrees
DRANG	F10.0	Deadrise angle in degrees

Example: DRAFT = 10. ft

DENS = 2. slugs/ft^3

BTANG = 0. deg

DRANG = 3. deg

1	10	20	30	40	50	60	70 80
10.	2.	0.	3.				

Cards 2, 3, and 4:

1	10	20	30	40	50	60	70 80
CØEF(1,1) (1,2) (1,3)	(2,1) (2,2) (2,3)	(3,1) (3,2) (3,3)	(4,1) (4,2) (4,3)	e	t	c.	(8,1) (8,2) (8,3)

Field: *Format* *Contents*

CØEF(8,3) 3(8F10.0) Polynomial Orders and Coefficients

Example:

$$K_1 = 0.32 + 8.5466395\ \xi + 150.46086\ \xi^2 \quad \text{for} \quad 0 \leq \xi < 0.0383972$$
$$K_2 = 2.1820894 - \cdots - 19339.04\ \xi^5 \quad \text{for} \quad 0.0383972 \leq \xi < 0.191986$$
$$K_3 = 4.748742 - \cdots - 1975.052\ \xi^5 \quad \text{for} \quad 0.191986 \leq \xi < 0.349066$$

1	10	20	30	40	50	60	70 80
2.	.32	8.5466395	150.46086	0.	0.	0.	0.
5.	2.1820894	-54.154911	668.88525	-4399.3721	14632.763	-19339.040	0.
5.	4.748742	-77.064451	517.53967	-1748.7944	2947.4596	-1975.0520	0.

Cards 5 and 6:

1	10	20	30	40	50	60	70 80
WIDTH(1) (9)	(2) (10)	(3)	(4)	(5)	(6)	(7)	(8)

Field: *Format*

WIDTH(10) 2(8F10.0) Width in feet of the slam region at Stations 11 through 20

Example: b=WIDTH=46 ft

1 10 20 30 40 50 60 70 80

46.	46.	46.	46.	46.	46.	46.	46.

1 10 20 30 40 50 60 70 80

46.	46.						

Card 7:

1 10 20 30 40 50 60 70 80

BSLP	I1281						

Field: *Format* *Contents*

BSLP BSLP is a parameter used by an iteration that determines the length of the slamming region. BSLP should be chosen so that

$$BSLP \geq \frac{1.1}{\Delta X} \max \{b_i - b_{i+1}\} \tan \beta + 0.1$$

where β is the deadrise angle, b_i, b_{i+1} are the half widths at adjacent stations, and ΔX is distance between stations

I1281 I10 Number of coefficients used to smooth the slam forces

Example: BSLP=0.1

1 10 20 30 40 50 60 70 80

.1	1281						

ΔΔΔ2. Hydrodynamic Force Data

<u>HYDRØ</u> - Hydrodynamic Force Calculation Card

1 10 20 30 40 50 60 70 80

HYDRØ	ISTN	ADDMS	BUØY	SCF	HDMP	YH	

Field: *Contents*

ISTN Station number at which HYDRØ forces are to be calculated. At present, ISTN can only assume values 3, 5, 7, 9, 11, 13, 15, 17, and 19.

ADDMS Linear added mass in ton-seconds squared per square foot ($\text{ton-s}^2/\text{ft}^2$) at station ISTN

Field: *Contents*

BUØY — Linear buoyancy in tons per square foot (ton/ft^2) at station ISTN

SCF — Smith correction factor in square feet per ton-seconds squared (ft^2/ton-s^2) at station ISTN

HDMP — Hydrodamping in ton-seconds per foot (ton-s/ft) at station ISTN

YH — Initial vertical displacement in feet at station ISTN. (This field is ignored if appropriate <u>INITIAL</u> cards are present in the input deck.)

Example: Station=ISTN=3

m_0=ADDMS=1.27805 ton-s^2/ft^2

k_b=BUØY=2.605691 ton/ft^2

ρA_0=SCF=0.9077 ft^2/ton-s^2

$C(\omega)$=HDMP=0.9 ton-s/ft^2

Y=YH=0

1	10	20	30	40	50	60	70 80
HYDRØ	3	1.27805	2.605691	.9077	.9		

<u>NL_A/NL_B</u> - Nonlinear Added Mass/Buoyancy Card

(a) Unconditional (No Restriction on Y_r and V_r):

1	10	20	30	40	50	60	70 80
NLA / NLB	ISTN	NEXP	C1	C2	C3	C4	

(b) Y_r Conditional:

1	10	20	30	40	50	60	70 80
NL1A / NL1B	ISTN	YC	ignored				+abc

1 10 20 30 40 50 60 70 80

+abc	CØDEY$_1$	NEXP$_1$	C1$_1$	C2$_1$	C3$_1$	C4$_1$	+def
+def	CØDEY$_2$	NEXP$_2$	C1$_2$	C2$_2$	C3$_2$	C4$_2$	

(c) Y_r and V_r Conditional:

1 10 20 30 40 50 60 70 80

<u>NL2A</u> <u>NL2B</u>	ISTN	YC	VC	ignored			+ghi
+ghi	CØDE2$_1$	NEXP$_1$	C1$_1$	C2$_1$	C3$_1$	C4$_1$	+jkl
+jkl	CØDE2$_2$	NEXP$_2$	C1$_2$	C2$_2$	C3$_2$	C4$_2$	+mno
+mno	CØDE2$_3$	NEXP$_3$	C1$_3$	C2$_3$	C3$_3$	C4$_3$	+pqr
+pqr	CØDE2$_4$	NEXP$_4$	C1$_4$	C2$_4$	C3$_4$	C4$_4$	

Field:	*Contents*
ISTN	Station number as before
YC	Y_c (Y_r - critical) in feet
VC	V_c (V_r - critical) in feet
CØDEY	Alphanumeric condition code (Y_r - conditional): Y<u>L</u>YC for $Y_r < Y_c$ Y<u>G</u>YC for $Y_r > Y_c$
CØDE2	Alphanumeric condition code (Y_r, V_r - conditional): Y<u>L</u>YC/V<u>L</u>VC for $Y_r < Y_c$, $V_r < V_c$ Y<u>G</u>YC/V<u>L</u>VC for $Y_r > Y_c$, $V_r < V_c$ Y<u>L</u>YC/V<u>G</u>VC for $Y_r < Y_c$, $V_r > V_c$ Y<u>G</u>YC/V<u>G</u>VC for $Y_r > Y_c$, $V_r > V_c$
C_s	Coefficients
NEXP	Expression number

NEXP	Expression	Number of Coefficients
1	C_1	1
2	$C_1 Y_r$	1
3	$C_1(Y_r - Y_c)$	1
4	$C_1 Y_r + C_2$	2
5	$C_1 Y_r^2$	1
6	$C_1 (Y_r - Y_c)^2$	1
7	$C_1 Y_r^2 + C_2 Y_r$	2
8	$C_1 Y_r^2 + C_2 Y_r + C_3$	3
9	$C_1 Y_r^3$	1
10	$C_1 (Y_r - Y_c)^3$	1
11	$C_1 Y_r^3 + C_2 Y_r^2$	2
12	$C_1 Y_r^3 + C_2 Y_r^2 + C_3 Y_r$	3
13	$C_1 Y_r^3 + C_2 Y_r^2 + C_3 Y_r + C_4$	4
14	$C_1 Y_r^2 + C_2$	2
15	$C_1 Y_r^3 + C_2 Y_r$	2
16	$C_1 Y_r^3 + C_2 Y_r + C_3$	3
17	$C_1 Y_r^3 + C_2$	2
18	$C_1 Y_r^3 + C_2 Y_r^2 + C_3$	3

Example: Station 17, Added mass 10^3 ton-s^2/ft^2

$$\begin{aligned}
\overline{m} &= -13.17\ Y_r + 1.568\ Y_r^2 && Y_r < 0,\ V_r > 0 \\
&= -13.17\ Y_r && > 0,\ > 0 \\
&= -13.17\ Y_r + 2.720\ Y_r^2 && < 0,\ < 0 \\
&= -13.17\ Y_r && > 0,\ < 0
\end{aligned}$$

NL2ADDMASS	17	0.	0.				DD1

+D1	LG	7	1.568E-3	-13.17E-3			DD2

+D2	GG	2	-13.17E-3				DD3

+D3	LL	7	2.72E-3	-13.17E-3			DD4

+D4	GL	2	-13.17E-3				

ΔΔΔ3. Ship Response Data

DAMPC - Calculated Structural Damping Card

1	10	20	30	40	50	60	70	80
DAMPC	CPSNAT	PCCP	Ignored					

Field: *Contents*

CPSNAT Ship natural frequency in hertz

PCCP Percent of critical damping, e.g., 10 percent is entered as .10

Example: Ship natural frequency=CPSNAT=1 Hz

Percent critical damping=2 percent

1	10	20	30	40	50	60	70	80
DAMP	1.	.02						

HALF-STTN - Half-Station Card

1	10	20	30	40	50	60	70	80
HALF-STTN	HSTN	SHPMS	EI	DAMPC	YDØT	BDM		

Field:	*Contents*
HSTN	Half-station number, e.g., 4.5
SHPMS	Ship mass per unit length in ton-s^2/ft^2 at half-station HSTN
EI	Bending rigidity in ton-feet squared (ton-ft^2) at half-station HSTN
DAMPC	Structural damping coefficient in ton-seconds per square foot (ton-s/ft^2) at half-station HSTN (This field is ignored if a DAMPC card is present in the input deck.)
YDØT*	Initial vertical velocity of ship in feet per second at half station HSTN.
BDM*	Initial bending moment in feet-tons at half station HSTN.

Example: At Station 1.5

HSTN = 1.5

m = SHPMS= 0.74401 ton-s^2/ft^2

EI = 7.76725×10^9 ton-ft^2

C = DAMPC= 0.080882

$\dot{Y}_0$ = YDØT = 0 (initial condition)

M_0 = BDM = 0 (initial condition)

1	10	20	30	40	50	60	70 80
HALF-STTN	1.5	.74401	7.76725+9	.080882			

STATIØN - Station Card

1	10	20	30	40	50	60	70 80
STATIØN	ISTN	KAG	RINT	GDØT	SF	ignored	

Field:	*Contents*
ISTN	Station number
KAG	Shear stiffness in tons at station ISTN

*Ignored if appropriate INITIAL cards are present in input deck.

Field:	*Contents*
RINT	Moment of inertia in ton-seconds squared at station ISTN
GDØT*	Initial angular velocity in radians per second at station ISTN
SF*	Initial shear force in tons at station ISTN

Example: ISTN = Station 0

$$KAG = 13.3987 \times 10^5 \text{ tons}$$

$$I_{mz} = RINT = 0.02686 \times 10^3 \text{ ton-s}^2$$

$$\dot{\gamma} = GD\emptyset T = 0 \text{ (Initial condition)}$$

$$V = SF = 0 \text{ (Initial condition)}$$

1	10	20	30	40	50	60	70	80
STATIØN	0	1339870.	26.86					

INITIAL - Initial Conditions Card

(Initial conditions cards are the punched final conditions of a previous run.)

1	10	20	30	50	70	80	
INITIAL	1	ignored	TSTRT	ignored			see INTGRTN card
INITIAL	2	HSTN	YDØT	BDM			see HALF-STTN card
INITIAL	3	ISTN	GDØT	SF			see STATIØN card
INITIAL	4	ISTN	YH	ignored			see HYDRØ card

SCTN LNGTH - Section Length Card

1	10	20	30	40	50	60	70	80
SCTN LNGTH	DXI	ignored						

*Ignored if appropriate INITIAL cards are present in input deck.

Field:	*Contents*
DXI	Distance between adjacent stations in feet

Example: ΔX = DXI = 41 ft

1	10	20	70	80
SCTN LNGTH	41.			

SPEED - Speed Card

1	10	20	70	80
SPEED	SPEED	ignored		

Field:	*Contents*
SPEED	Ship speed in feet per second

Example: U = SPEED = 27 fps

1	10	20	70	80
SPEED	27.			

COMMENT - Comment Card

1	10	70	80
CØMMENT	CØMMENT		

Field:	*Contents*
CØMMENT	Commentary material inserted into echo listing of input deck. The CØMMENT cards are otherwise ignored.

Example:

1	10	70	80
CØMMENT	ESSEX DISCRETE WAVE TRAIN		

ΔΔΔ4. Option Selection Data

CØNTRØL - Control Card

1	10	20	30	40	50	60	70 80
CØNTRØL	ignored	$CNTRL_2$	ignored	$CNTRL_3$	ignored	$CNTRL_4$	

Field: *Contents*

$CNTRL_2$	Punch YES if having final conditions punched
$CNTRL_3$	Punch YES if including nonlinear hydrodynamic force
$CNTRL_4$	Punch YES if including hydrodynamic damping

Example:

1	10	20	30	40	50	60	70 80
CØNTRØL	PUNCH=	YES	NL HYDRØ=	YES	HYD DAMP=	YES	

DEBUG - Debug Card

1	10	20	30 70	80
DEBUG	IBEG	IEND	ignored	

Field: *Contents*

IBEG	Intermediate calculations will be put out between derivative evaluations number IBEG
IEND	IEND

Example: In the subroutine KUTMER, HCX is the smallest step size used in the integration. For debugging, it has been decided to print,

first, two steps for examination; then, IBEG=1 and IEND=2 and debug cards will be as follows

1	10	20	30	70	80
DEBUG	1	2			

INTGRTN - Integration Card

1	10	20	30	40	50	60	70	80
INTGRTN	HMNI	TSTRT	TF	TCPU	ignored		+aaa	

+aaa	YHE	YDØTE	GDØTE	BDME	SFE	ignored	+bbb

+bbb	YHAB	YDØTAB	GDØTAB	BDMAB	SFAB	ignored	

Field: *Contents*

HMNI	Maximum internal step size
TSTRT	Start of simulation in simulation seconds. This is ignored if an appropriate INITIAL card is present in input deck.
TF	End of simulation in simulation seconds
TCPU	Job time limit in CPU seconds

Max Relative Error in Units Per Unit	Max Absolute Error in Unit	Unit	Dependent Variable
YHE	YHAB	feet	Vertical displacement
YDØTE	YDØTAB	feet per second	Vertical velocity
GDØTE	GDØTAB	radians per second	Angular velocity
BDME	BDMAB	foot-tons	Bending moment
SFE	SFAB	tons	Shear force

Example:

HMNI = 0.0125 s
TSTRT = 0 s
TF = 2 s
TCPU = 1500 s
YHE = 0.1 ft
YHAB = 1 ft
YDØTE = 0.01 ft/s
YDØTAB = 1 ft/s
GDØTE = 0.01 rad/s
GDØTAB = 0.2 rad/s
BDME = 0.2 ft-ton
BDMAB = 2000 ft-tons
SFE = 0.05 ton
SFAB = 300 tons

1	10	20	30	40	50	60	70 80
INTGRTN	.0125	0.	2.	1500.			+INT1

+INT1	.1	.01	.01	.2	.05		+INT2

+INT2	1.	1.	.2	2000.	300.		

ΔΔΔ5. **Output Description Data**

AXES - Axes Card for Plotting

1	10	20	30	40	50	60	70 80
AXES	P_ℓ	P_u	B_ℓ	B_u	W_ℓ	W_u	

Axis Lower Limit	Axis Upper Limit	Unit	Plotted Variable
P_ℓ	P_u	degree	Pitch
B_ℓ	B_u	foot-ton	Midship bending moment
W_ℓ	W_u	feet	Wave height at bow

Example: $P_\ell = -3$ deg $\quad P_u = 3$ deg

$B_\ell = -6(10)^5$ ft-ton $\quad B_u = 6(10)^5$ ft-ton

$W_\ell = -12$ ft $\quad W_u = 12$ ft

<u>PLØT</u> - Plot Card

1	10	20	30	40	50	60	70 80
<u>PLØT</u>	NAME	CØDE	PHØNE	W_1	R	ignored	

Field: *Contents*

NAME, CØDE, PHONE — PLOT IDENTIFICATION

Field: *Contents*

W_1 Time in simulation seconds between plotted points

R Number of simulation seconds of data per frame

Example: Name = Schroeder

Code = 1844

Phone = 71426

W_1 = 0.1 s between plotted points

R = 60 s per frame

1	10	20	30	40	50	60	70 80
PLØT	SCHROEDER	1844	71426	0.1	60.		

PRINT - Print Card

1 PRINT	10 TIMES	70	80

Field: *Contents*

TIMES — Print times in simulation seconds expressed in free-field "bias width" notation. For example, 0(.1) 1(.5) 5(.1) 6$ represents 0, 0.1, 0.2, . . . 0.9, 1, 1.5, 2, 2.5, . . ., 4.5, 5, 5.1, 5.2, . . ., 5.9, 6, and 6$ means that 6 is last simulated second.

Example: Print results every second from 0 to 90 simulation seconds.

1 PRINT	10 0. (1.) 90. $	70	80

ΔΔΔ6. Sea Dynamics Data

SEA-GEN - Sea Generator Card (Case 1 - Point application of sinusoidal standing wave to determine ship vibratory frequency.)

1	10	20	30	70	80
SEA-GEN	FREQ	WVHGH	ignored		

Field: *Contents*

FREQ — Wave frequency in hertz

WVHGH — Wave height in feet

SEA-GEN - Sea Generator Card (Case 2 - Sinusoidal sea wave application with wave velocity $C = g/\omega$ to determine ship RAO's.)

1	10	20	30	70	80
SEA-GEN	FREQ	WVHGH			

SEA-DATA - Sea Generator Card (Case 3 - Discrete wave train; at present, program is written for ESSEX only; it will be necessary to revise the program for sea trial data of other ships.)

1	10	20	30	40	50	60	70 80
SEA-DATA	OMEGA(1)	OMEGA(2)	OMEGA(3)	TAU(1)	TAU(2)	TAU(3)	

Example: ω_1 = OMEGA (1) = 1.08470672 rad/s

ω_2 = OMEGA (2) = 0.598444 rad/s

ω_3 = OMEGA (3) = 0.804804 rad/s

τ_1 = TAU (1) = 4.11238748 rad

τ_2 = TAU (2) = -21.97272 rad

τ_3 = TAU (3) = -29.54952 rad

1	10	20	30	40	50	60	70 80
SEA-DATA	1.08470672	.598444	.804804	4.11238748	-21.97272	-29.54952	

REFERENCES

1. Andrews, J.N. and S.-L. Chuang, "Seaworthiness Analog Computer," David Taylor Model Basin Report 1829 (Aug 1965).

2. Jasper, N.H. and J.T. Birmingham, "Strains and Motions of USS ESSEX (CVA-9) during Storms Near Cape Horn," David Taylor Model Basin Report 1216 (Aug 1958).

3. Andrews, J.N., "A Method for Computing the Response of a Ship to a Transient Force," David Taylor Model Basin Report 1544 (Nov 1963).

4. McGoldrick, R.T., "Ship Vibration," David Taylor Model Basin Report 1451; Figure 3-8, p. 3-18; Figures A-2 and A-3, p. A-3; Table 8-2, p. 8-10 (Dec 1960).

5. Chuang, S.-L. et al., "Experimental Investigation of Catamaran Cross-Structure Slamming," NSRDC Report 4653 (Sep 1975).

6. Henderson, F., "Transient Response Calculation in the Frequency Domain with General Bending Reponse Program (GBRP)," NSRDC Report 3613 (Feb 1971).

7. Kaplan, P. et al., "An Investigation of the Utility of Computer Simulation to Predict Ship Structural Response in Waves," Ship Structure Committee Report SSC-197 (Jun 1969).

8. Bruck, H.A., "Procedure for Calculating Vibration Parameters of Surface Ships," NSRDC Report 2875 (Oct 1968).

9. Carnahan, B. et al., "Applied Numerical Methods," Chapter 1, John Wiley & Son Inc., New York (1969).

10. Grim, O., "Berechnung der durch Schwingungen eines Schiffskörpers erzeugten hydrodynamischen Kräfte," Jahrbuch der Schiffbautechnischen Gesellschaft, Vol. 47 (1953).

11. Golovato, P., "A Study of the Forces and Moments on a Heaving Surface Ship," David Taylor Model Basin Report 1074 (Sep 1957).

INITIAL DISTRIBUTION

Copies		
7	CHONR	
	1 ONR 400R	
	1 ONR 400R1	
	1 ONR 430B	
	1 ONR 432	
	1 ONR 436	
	1 ONR 474	
	1 ONR 438	
1	CHONR, London	
2	NAVMAT	
	1 MAT 03T2	
	1 MAT 034	
3	NRL	
	1 2620	
	1 4000	
	1 8400	
1	USNA	
1	USNAVPGSCHOL	
1	NWC, China Lake	
10	NAVSEA	
	1 SEA 00G	
	1 SEA 012	
	1 SEA 03	
	1 SEA 031	
	1 SEA 032	
	1 SEA 0322	
	1 SEA 035	G. Sorkin
	1 SEA 0351	C. Pohler
	2 SEA 09G3	
3	NAVAIR	
	1 5301	
	1 5302	
	1 520	
2	NAVFAC	
	1 FAC 03	
	1 FAC 04	

Copies		
18	NAVSEC	
	1 SEC 6105F	D. Lewis
	1 SEC 6110	
	1 SEC 6113	
	1 SEC 6113B5	J. Berner
	1 SEC 6114	
	2 SEC 6114D	E. Aronne
		T. Wickert
	1 SEC 6120	
	1 SEC 6120D	
	1 SEC 6122	
	4 SEC 6128	S. Arntson
		J. O'Brien
	4 SEC 6136	R. Johnson
		E. Comstock
		D. Byers
12	DDC	
2	USCG, Washington	
	1 SSC	W. Benkert
1	Lib of Congress	
1	MARAD	
2	COMSC	
	1 M4E	
	1 M4E4	
1	Nat'l Sci Foundation	
2	Univ. of Calif, Dept of NA	
	1 Library	
	1 Mr. O. J. Sibul	
1	Catholic Univ.	
	1 Prof. M. Soteriades	
1	MIT, Dr. A.H. Keil	
1	MIT, Dept. Ocean Engr./ Dr. N. Jones	
2	Univ. of Michigan, Dept of NAME	
	1 Prof. H. Benford	
	1 Prof. F. Ogilvie	

PRECEDING PAGE BLANK-NOT FILMED

Copies	
2	SIT, Davidson Lab 1 Mr. John Dalzell 1 Library
1	SWRI
1	Web Institute, E.V. Lewis
3	SNAME 1 Slamming Panel 1 Hull Loading Panel
3	ABS 1 Dr. A. Elbatouti 1 Mr. Stiansen 1 Library
1	Design Assoc., Inc./ Mr. M. Kawasaki
1	GIANNOTTI & BUCK ASSOC., INC./ Dr. J. G. Giannotti
1	J. G. ENGR. Res Assoc./ Dr. J. E. Greenspan
1	Dr. Norman H. Jasper
1	Nashville Bridge Co./ Mr. K. E. Hoffmeister
1	National Maritime Res Cen/ Dr. W. M. Maclean
1	Oceanics, Inc./ Mr. P. Kaplan
1	Wheeler Industries, Inc./ Mr. Edgar D. Hoyt

CENTER DISTRIBUTION

Copies	Code	
1	15	
1	1506	
1	152	
1	154	
1	156	
1	1568	
1	1572	
1	17	
1	172	
1	173	
1	1730.3	
1	1730.3	J. Andrews
30	1730.3	S. Chuang
1	1730.5	
1	174	
1	177	
1	178	
1	18	
1	1809	
1	184	
1	1805	E. Cuthill
1	1844	S. Dhir
5	1844	E. Schroeder
5	1844	S. Wybraniec
4	1962	G. Antonides N. Huzil A. Kukk A. Hagen
30	5214.1	Reports Distribution
1	522.1	Library C
1	522.2	Library A